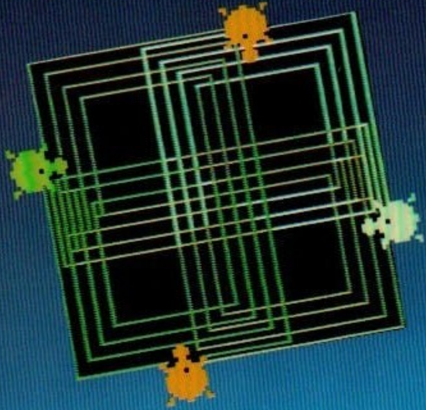
PHILIPS

De logische manier om met een computer om te gaan



MSX-LOGO

Created by Logo Computer Systems Inc.

MSX™

MSX-LOGO

LCSI

PHILIPS

Een Inleiding tot MSX-LOGO

door
LCSI
Logo Computer Systems Inc.
Montreal, Canada H8T 1A1

LICENTIE VOOR EINDGEBRUIKERS VOLGENS LCSI-STANDAARDS

Licentie:

LICENTIEGEVER verleent de eindgebruiker een licentie om de programmatuur te gebruiken onder de volgende voorwaarden:

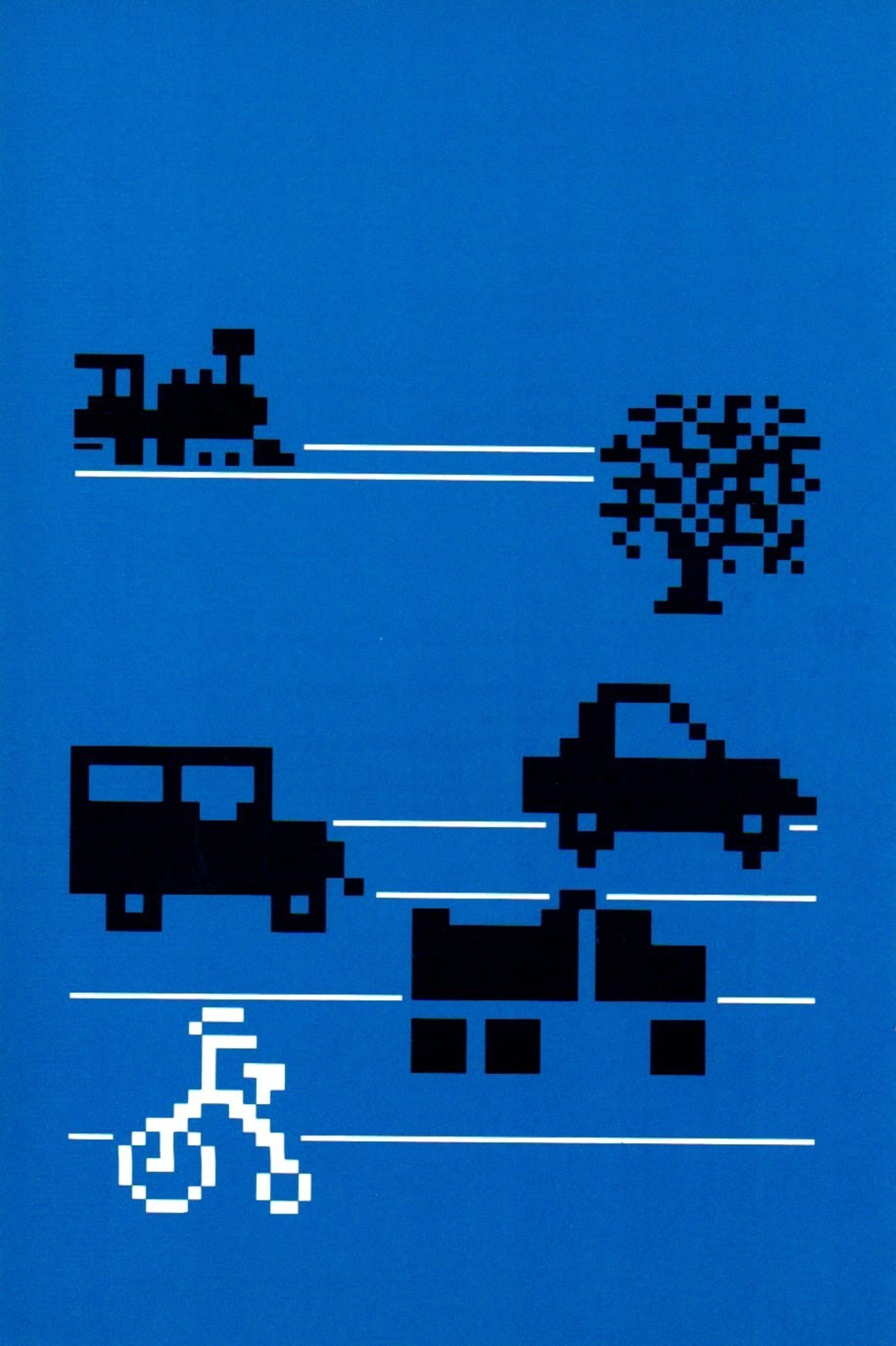
1. De eindgebruiker mag de programmatuur slechts op één computersysteem tegelijk gebruiken.
2. De eindgebruiker mag deze licentie en de programmatuur overdragen aan derden, indien:
   (a) De andere partij instemt met alle bepalingen die in deze overeenkomst zijn geformuleerd.
   (b) De eindgebruiker geen andere exemplaren van deze programmatuur in eigen bezit houdt.

Met dank aan

Schildpadtaal:	
auteur:	Seymour Papert
mede-redactrices:	Judy Le Gallais
	Barbara Mingie
tekstverwerking:	Lise van Chestein
Het Beknopt Naslagwerk:	
mede-redactrices:	Judy Le Gallais
	Barbara Mingie
tekstverwerking:	Lise van Chestein
Grafische presentatie en opmaak:	Julien Perron
	Richard Lavigne
MSX-Logo software team:	
analisten:	Mario Bergeron
	Claude Chouinard
	Mamadou Billo Diallo
	Brian Silverman
tester:	Paul Legault
Speciale medewerking:	Eric Brown
	Sharnee Chait
	Susan Fischer
	Jenny Sparrow
Project management:	Effie Maniatis

Inhoud

Het Beknopt Naslagwerk



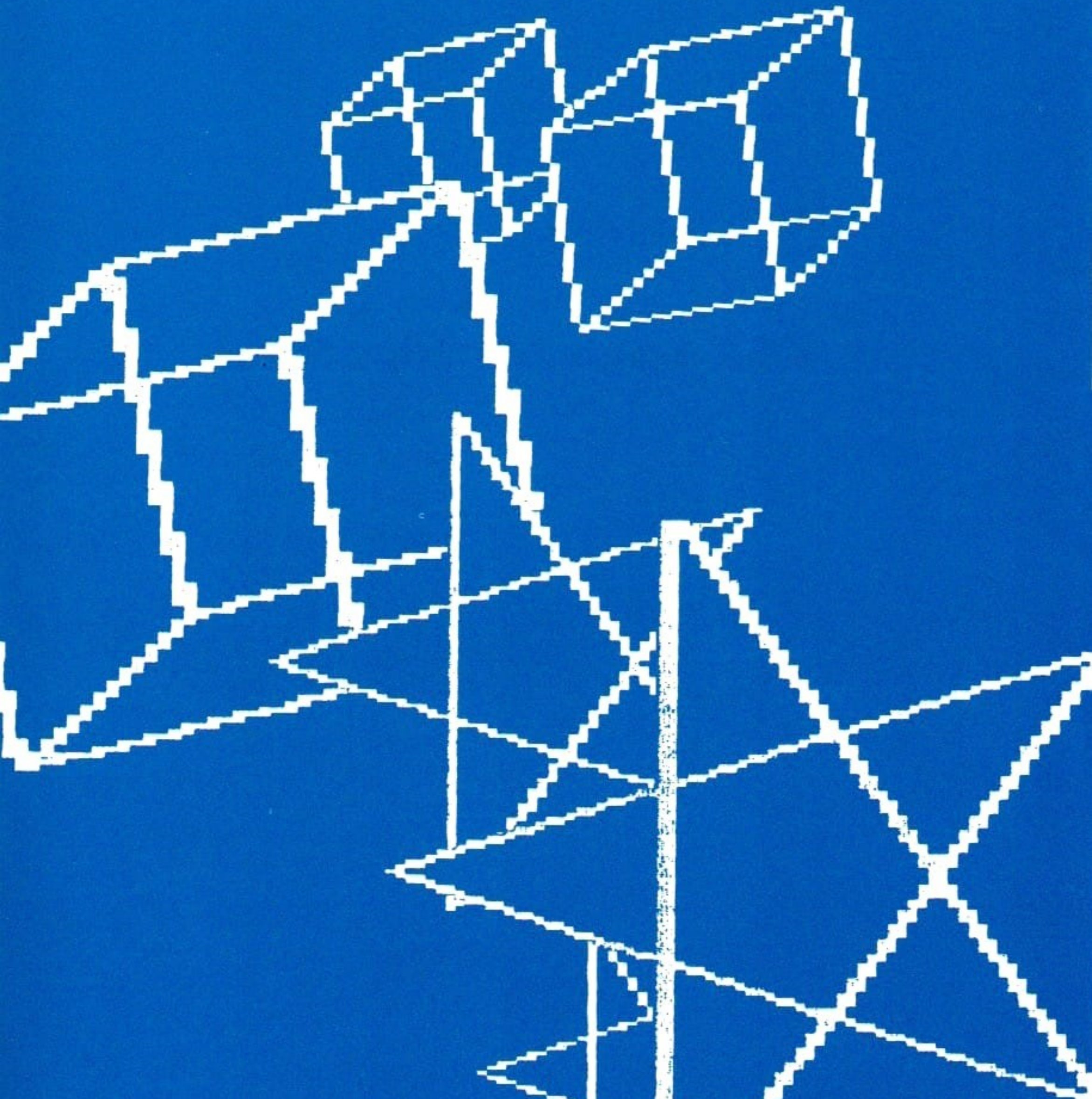
Voorwoord

Een inleiding tot MSX-Logo is verdeeld in twee gedeelten. Het eerste deel, *MSX-Logo Schildpadtaal,* is een inleiding op het programmeren in Logo. In dit deel lees je hoe je de schildpad opdrachten geeft, je eigen "programma's" schrijft en je werk opslaat op een schijf of cassettebandje. Je vindt in dit gedeelte ook voorbeelden van Logo-programma's die je kunt schrijven.

In het tweede deel van het boek, *Het Beknopt Naslagwerk,* vind je meer gedetailleerde informatie over MSX-Logo, zoals het gebruik van de kleuren, de manieren waarop je de schildpad kunt veranderen en de demonen; functies om tekst en vormen te editen; speciale toetsen; speciale karakters; een beschrijving van de manier waarop iedere primitieve procedure werkt; en een lijst van de primitieve procedures, ingedeeld in functiegroepen.

Je zult in het boek tekst tegenkomen in blauwe en zwarte kleur. `Blauwe` tekst wordt gebruikt om aan te geven wat je zelf moet typen op je computer. De zwarte tekst geeft aan wat de computer op het beeldscherm weergeeft.

MSX-Logo Schildpadtaal

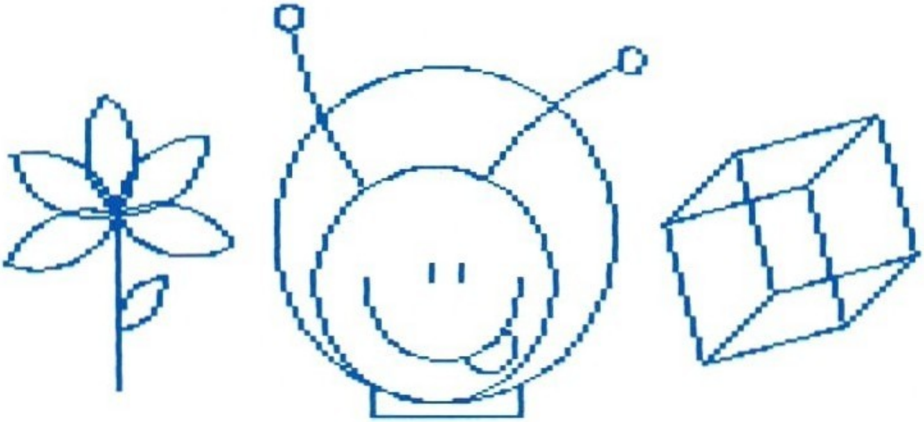


Hoofdstuk 1

Inleiding: Welkom bij MSX-Logo

Programmeren wil zeggen dat je de computer vertelt wat hij moet doen. Omdat computers geen Nederlands verstaan, zul je een taal moeten gebruiken die de computer kan verstaan, maar die wij mensen ook kunnen verstaan. Logo is zo'n taal, een programmeertaal. Logo vertelt de computer wat jij wil dat de computer moet doen. Hiervoor gebruiken we speciale woorden, die we primitieven of ook wel *primitieve procedures* noemen. Primitieve procedures worden gebruikt om instructies te maken die Logo samen met de computer uitvoert.

Je hebt ongetwijfeld wel eens "computer graphics" gezien; dat zijn tekeningen die door computers zijn gemaakt. Nog even, en je zult ze zelf ook maken met je computer. Blader maar door dit boek als je er nog meer voorbeelden van wilt zien.

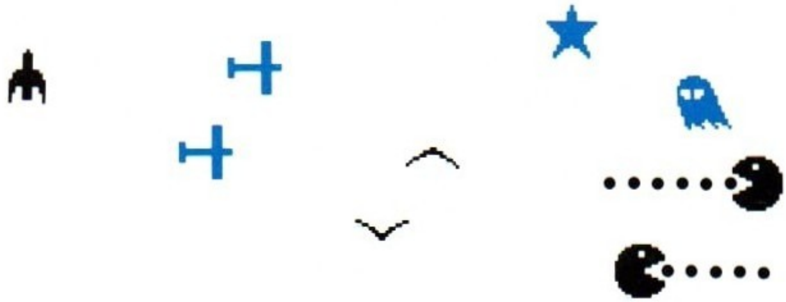


Heb je wel eens met een computer gewerkt die jou vragen stelde en dan je antwoorden controleerde?

`5 + 7 =`	Vraagt de computer.
`57`	Typ je als antwoord.
`Fout! Probeer het nog eens`	Antwoordt de computer.

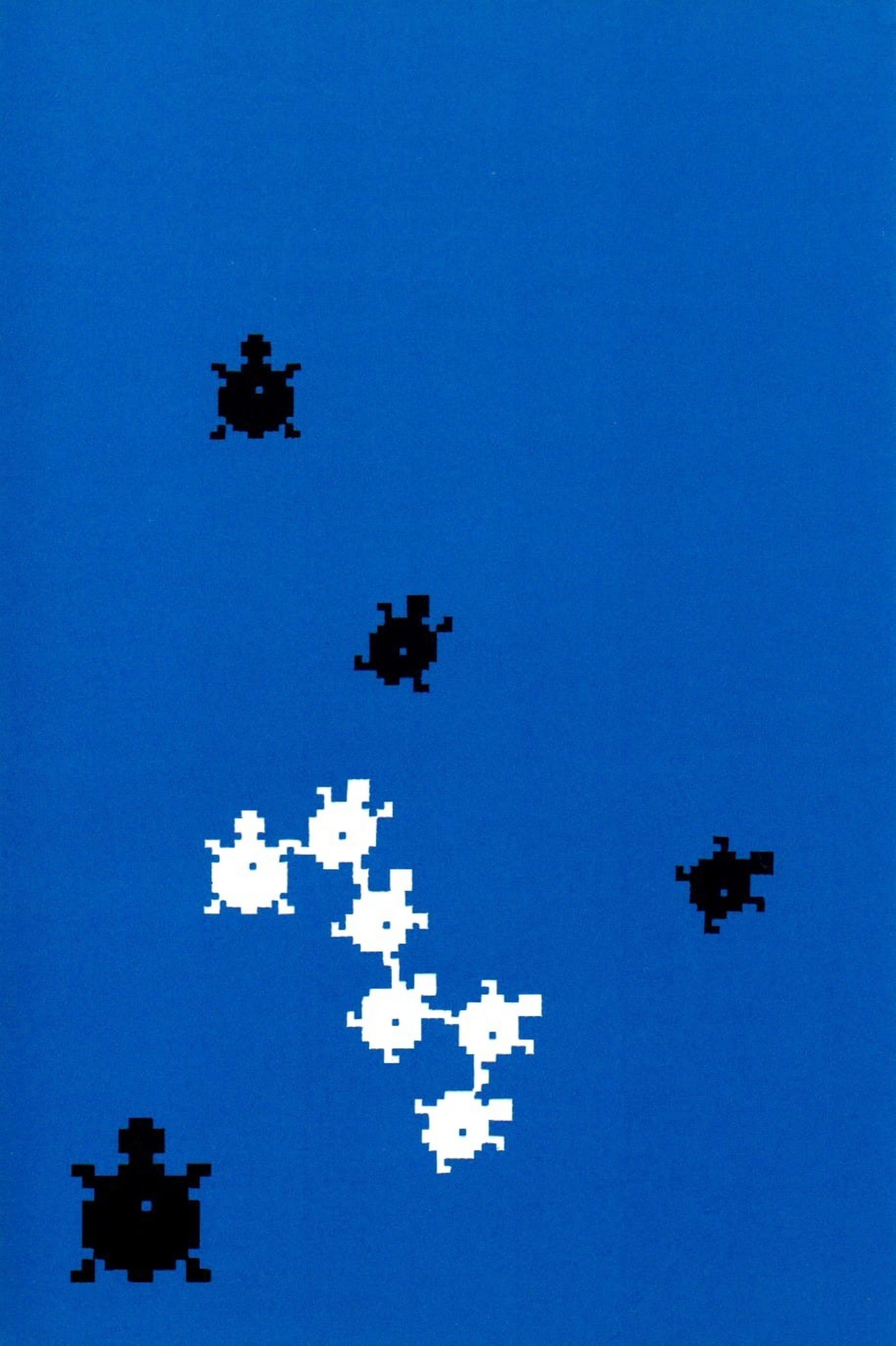
Als je wilt, kun je zelf leren hoe je je computer programmeert, zodat hij op deze manier werkt.

Voor ieder videospel is een programmeur aan het werk geweest die de computer heeft verteld hoe hij de monsters moet tekenen, hoe hij ze moet laten bewegen, enzovoort. Jij kunt ook leren om zelf je eigen videospellen te programmeren.





En dit is nog maar het begin. Het meeste plezier zul je beleven aan het bedenken van je eigen tekeningen. Met andere woorden, programmeren wat nog nooit eerder geprogrammeerd is.



Hoofdstuk 2

Hoe Je Begint

1. Steek de Logo-cassette in je machine.
2. Als je een schijfeenheid of een data recorder op je machine hebt aangesloten, moet je die aan zetten.
3. Zet de monitor of het t.v.-toestel aan.
4. Zet nu de computer aan.

Let op: als je een schijfeenheid of een data recorder wil gebruiken, moet je eraan denken dat je die eerst aansluit op de computer voordat je de computer aan zet.

Ontmoet de Schildpad

Een reis van duizenden kilometers begint met één stap. Onze eerste stap naar het schrijven van Logo-programma's bestaat uit een ontmoeting met de schildpad.

De schildpad is een computerwezen, dat alleen maar op het beeldscherm leeft. Hij maakt tekeningen als je hem daarvoor de juiste instructies geeft. Door te typen, kun je met de schildpad praten. Probeer dit maar eens:

```
?vooruit 5Ø
```

Vergeet niet de spatie te typen.

In de computerwereld schrijven we een nul zo: Ø
Ø is het getal nul. O is de letter O.

Als je een Frans AZERTY-toetsenbord gebruikt en je wil cijfers typen, dan doe je dat op de volgende manier: je houdt de **SHIFT** -toets ingedrukt en drukt dan op de cijfertoets met het cijfer dat je wilt gebruiken.

Wat gebeurt er als ik een typefout maak?

Gebruik de toets **BS** *(***B***ACK* **S***PACE = spatie terug) om de typefout te verwijderen.*

Je ziet nu dit op het scherm:



Waarom is de schildpad niet vooruit gegaan?

Je moet op de **Return** *-toets drukken om Logo te vertellen dat je klaar bent met je instructie.*

Nu zie je dit:



Als dit niet werkt, ben je misschien vergeten op de **Return** -toets te drukken of heb je een **O** getypt in plaats van een **Ø.** Typ **vooruit 5Ø** en druk op de **Return** -toets. Je kunt **vooruit** ook afkorten tot **vt.** Dat is gemakkelijker en spaart tijd uit, maar betekent precies hetzelfde als vooruit.

Kijk nu ook wat er gebeurt als je de instructie **vooruit** typt, gevolgd door een ander getal, bijvoorbeeld **vooruit 25.**

We noemen dit een primitieve procedure. → `vooruit` ← En dit noemen we de input. → `25`

```
vooruit 25
```

Het geheel is een instructie.

MSX-Logo kan primitieve procedures alleen maar begrijpen als je ze in kleine letters typt. Gebruik dus nooit hoofdletters.

Een nieuwe primitieve procedure

```
?ss
```

Je hoeft hier geen input achter te typen, maar je moet wel op de **Return** -toets drukken.

Waarom geen input? Bij **vt** moet je input opgeven om aan te geven hoe ver de schildpad vooruit moet lopen. Maar **ss** betekent gewoon *scherm schoonmaken.* Meer hoef je dus niet te vertellen.

Wat gebeurt er als ik een fout maak?

Probeer maar.

Typ geen spatie en kijk wat er dan gebeurt:

?vt5Ø **[Return]**

Typ geen getal en kijk wat er gebeurt:

?vt **[Return]**

Beide keren zette Logo een bericht op het scherm. Deze berichten gaven aan wat je fout deed. Probeer het nog een keer.

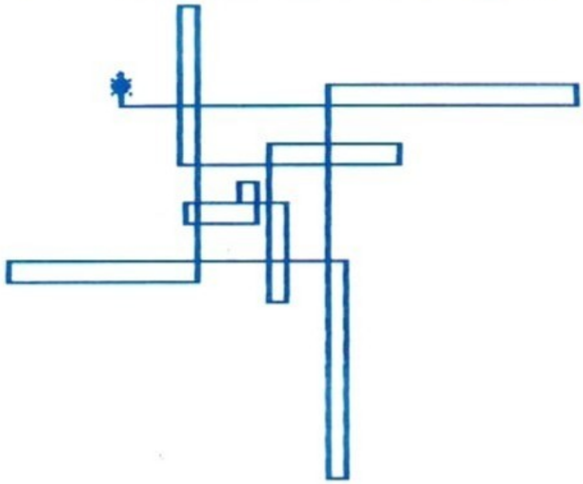
Je kunt de schildpad in een andere richting laten gaan door het volgende te typen:

`?rechts 9Ø` of: `?re 9Ø`

Spatie — Vergeet niet op **Return** te drukken.

Probeer nu **vt** ⎵ **re** ⎵ **vt** ⎵ **re** ⎵ enzovoort.

Het streepje ⎵ betekent dat je hier een getal mag invullen dat je zelf wilt. Als je een laag getal invult, loopt de schildpad een klein stukje verder. Bij grote getallen zal de schildpad je voor verrassingen stellen.





Probeer dit eens:



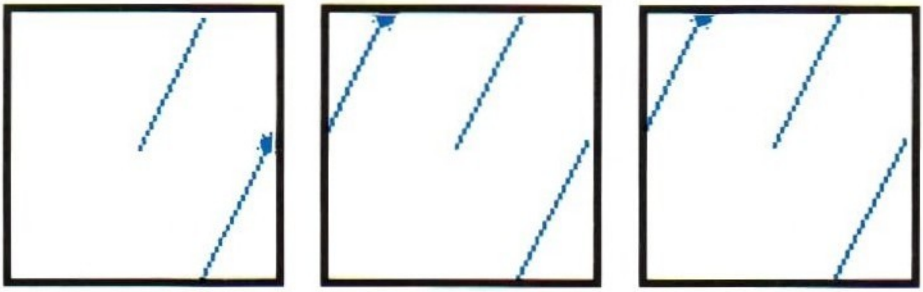
`?ss`



`?re 43`



`?vt 93`



?vt 90 ?vt 100 ?

Voor dit verschijnsel hebben we een naam. We zeggen dan dat de schildpad zich om het scherm heen *wikkelt.*

Probeer nu het volgende:

```
?ss
?re 5 vt ␣
```

Nog twee primitieve procedures

`achteruit ␣` of: `at ␣`

`links ␣` of: `li ␣`

De schildpad kan bewegen zonder dat hij lijnen tekent. Je hoeft hem dan alleen maar te vertellen dat hij zijn pen omhoog moet houden.

Op die manier kun je iets tekenen als:



Hier zijn de primitieve procedures die je moet gebruiken:

`penop` of: `po`

`penneer` of: `pn`

Probeer ze maar:

```
?ss re 90
?pn vt 10
?po vt 10
?pn vt 10
?po vt 10
```

Alle primitieve procedures die je tot nu toe hebt gezien, noemen we opdrachten. *Deze opdrachten vertellen Logo wat er moet gebeuren.*

Hier is een nieuwe primitieve procedure:

```
?herhaal 6 [pn vt 10 po vt 10]
```

Dit is een opdracht die twee inputs heeft.

De ene input is een getal.

De andere is een lijst met instructies: **[pn vt 1Ø po vt 1Ø]**

Zoals je ziet is de lijst ingesloten tussen vierkante haken [ ].

Als je een Frans AZERTY-toetsenbord gebruikt, dan gebruik je voor het typen van rechte haken drie toetsen. Als je **[** of **]** wilt typen, dan druk je eerst op **CODE** . Daarna moet je de **SHIFT** -toets ingedrukt houden en op **(** of **)** drukken. Tot slot moet je nog een keer op **CODE** drukken, zodat je deze toets weer uitschakelt.

Nog een paar opdrachten

`?thuis` Als je denkt dat er niets gebeurt, probeer dan dit maar eens:

```
?vt 5Ø re 9Ø vt 5Ø
?thuis
```

`?schoon` Probeer dit:

```
?pn vt 5Ø
?schoon
```

`?sw` **sw** staat voor schildpad weg.
Kijk maar of je de schildpad nog ziet!

`?ts` **ts** staat voor toon schildpad.

`?gum` Met een **gum** kun je **gum**men.
Probeer dit:

```
?pn vt 5Ø
?gum at 25
```

Als de schildpad weg is van je scherm, typ je **ts.**

Als de schildpad geen lijnen tekent, typ je **pn.**

Als je het scherm leeg wilt maken, typ je **ss.**

De Kracht van Logo Gebruiken

Hier zie je twee manieren om een vierkant te tekenen:

De omslachtige manier

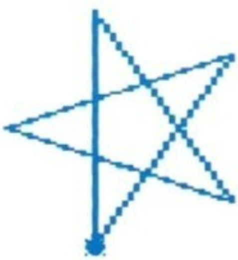
```
?vt 5Ø re 9Ø
?vt 5Ø re 9Ø
?vt 5Ø re 9Ø
?vt 5Ø re 9Ø
```

De korte manier

```
?herhaal 4 [vt 5Ø re 9Ø]
```

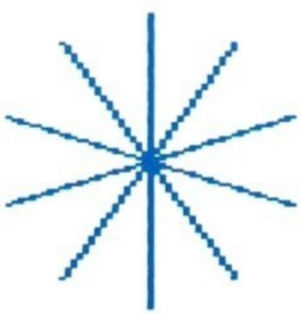
Met **herhaal** kun je veel plezier beleven. Wanneer je een andere input opgeeft, krijg je verbazingwekkende resultaten.

```
?herhaal 5 [vt 7Ø re 144]
```



Denk eraan: de lijst moet tussen vierkante haken staan, dus [ ], en niet tussen ronde haken.

```
?herhaal 1Ø [vt 5Ø at 5Ø re 3→
6]
```



Als je klaar bent met je werk aan het scherm kun je het beste eerst het scherm schoonmaken met **ss** *voordat je verder gaat.*



Hoofdstuk 3

Beweging, Vormen en Kleur

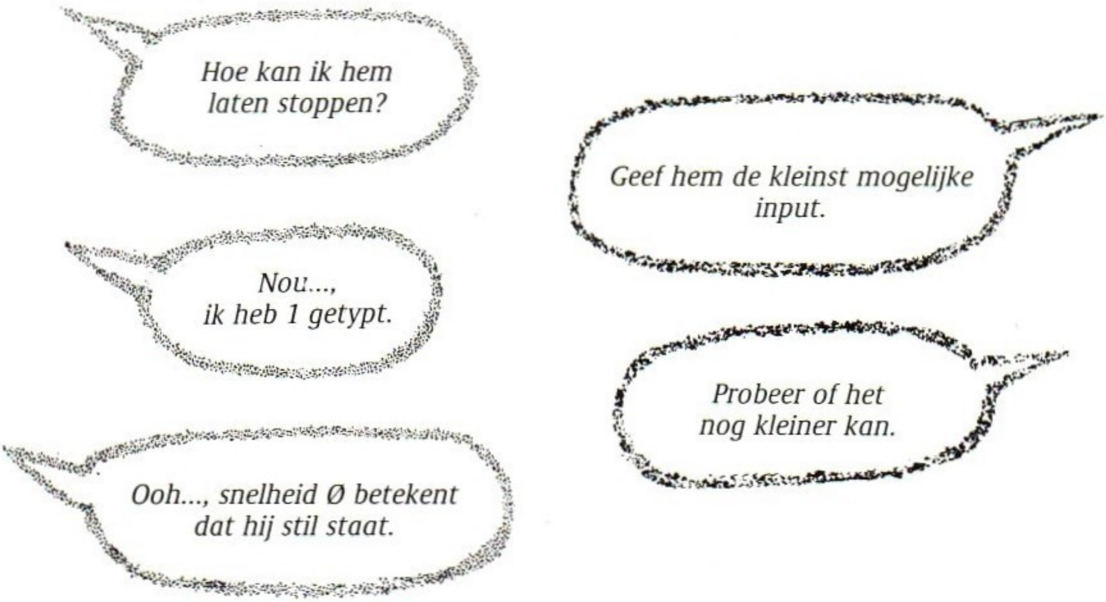
De Schildpad gaat lopen

?zetvaart 25 — **zetvaart** geeft de schildpad een bepaalde snelheid

> De schildpad moet nu bewegen. Zo niet, dan moet je het nog een keer proberen. Je typt dus **zetvaart 25** en drukt dan op de **Return** -toets.

Terwijl de schildpad loopt, kun je hem van richting laten veranderen met **re 9Ø** of **li 15** of met **re** en **li** met andere input. Kijk ook wat er gebeurt wanneer je **po**, **schoon** en **pn** typt als de schildpad aan het lopen is. Probeer daarna **vt 5Ø**.

Kijk hoe je de schildpad sneller en langzamer kunt laten gaan.



Vormen en Kleuren

Je kunt de vorm en de kleur van de schildpad veranderen. Als je hem een andere vorm geeft, verandert er in feite niets. Eigenlijk blijft het namelijk een schildpad.

`?zetv 5` **zetv** staat voor **zet v**orm.

De input is het nummer van de vorm die je kiest.

Iedere vorm die je de schildpad kunt geven, heeft in het geheugen van Logo een eigen nummer. Het eerste nummer is Ø.

Hoeveel verschillende vormen zijn er eigenlijk?

Nou! Tel zelf maar.

Kleuren hebben ook een nummer

`?kleursp 1` **kleursp** staat voor de **kleur** van de **s**child**p**ad.

`?kleursp 7`

Als je een televisie of monitor met een zwart/wit-scherm gebruikt, zien de kleuren eruit als verschillende tinten grijs.

De schildpad kan ook in verschillende kleuren tekenen. Probeer dit maar:

`?zetpk 9` **zetpk** staat voor **zet** de **p**en**k**leur.

en dan:

`?vt 3Ø`

Als je de schildpad zijn oorspronkelijke vorm wilt teruggeven, typ je:

```
?zetv 36
```

Als je de oorspronkelijke kleur van de schildpad terug wilt hebben, typ je:

```
?kleursp 15
```

Als je terug wilt gaan naar de oorspronkelijke kleur van de pen, typ je:

```
?zetpk 15
```

Meer dan één Schildpad

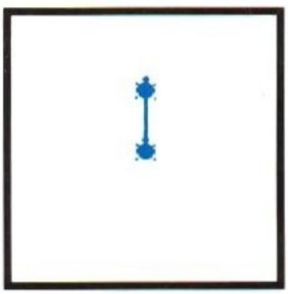
Je hebt tot nu toe nog maar één schildpad gezien. Maar er zijn er 3Ø. Elk van deze schildpadden heeft een eigen nummer. De schildpad die je inmiddels al kent, is schildpad Ø.

Als dat de eerste schildpad is, waarom is dat dan niet nummer 1?

Logo telt vanaf Ø, dus Ø, 1, 2, 3... in plaats van 1, 2, 3, 4...

Om schildpad 1 tevoorschijn te halen, typ je:

```
?zeg 1 ts
?vt 45
```



O ja, dat weet ik nog... **ts** *betekent* **t***oon* **s***childpad.*

Er staan nu twee schildpadden op je beeldscherm. Alle instructies die je vanaf nu geeft, gaan naar schildpad 1. Alleen deze schildpad luistert naar de instructies; maar schildpad Ø zal er niet op reageren.

Als je wilt dat schildpad Ø weer luistert, typ je:

```
?zeg Ø
```

Je kunt echter ook beide schildpadden laten luisteren. Dat doe je zo:

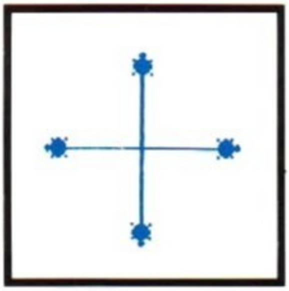
```
?zeg [Ø 1]
```

Vergeet de spatie niet.

Denk eraan dat dit een lijst is. Hij moet dus tussen vierkante haken staan, [ ].

Probeer dit eens:

```
?ss
?zeg 1 re 90
?zeg 2 re 90 + 90
?zeg 3 re 90 + 90 + 90
?zeg [Ø 1 2 3]
?ts zetvaart 25
```



Ze moeten nu allemaal bewegen.

Dit is een slimme manier om dit allemaal te herroepen en maar tegen één schildpad tegelijk te praten:

```
?zeg allemaal ss sw
```

De opdracht **sw** betekent **s**childpad **w**eg.

```
?zeg Ø ts pn
```

maar je kunt ook gewoon het volgende typen:

```
?ms
```

De opdracht **ms** betekent **m**aak **s**choon. Hiermee wordt alles op het scherm, dus alle schildpadden en kleuren, teruggezet naar de beginstand, toen je Logo startte.

Een Project

Schildpadden snellen over de straat:



Probeer eerst of je dit zelf kunt maken, en kijk *dan pas* hieronder of je hetzelfde idee had als wij.

Als je de instructies typt, vind je het misschien handig om zo nu en dan de tekst van je scherm weg te halen. Hiervoor gebruik je de opdracht **tw** (van **t**ekst **w**eg).

Schildpadtaal	Mensentaal
`?ms`	Haalt alles weg, laat schildpad Ø luisteren en maakt hem zichtbaar op het scherm.
`?po vt 16`	De schildpad gaat naar de bovenkant van de straat.
`?re 9Ø`	De schildpad draait naar rechts en is klaar om te gaan tekenen.
`?pn vt 256`	De bovenkant van de straat wordt getekend over de hele breedte van het beeldscherm.
`?li 9Ø po`	De schildpad draait zich om en kijkt nu weer omhoog.
`?at 32 re 9Ø`	De schildpad gaat omlaag naar de overkant van de straat.
`?pn vt 256`	De onderkant van de straat wordt getekend over de hele breedte van het beeldscherm.
`?po thuis`	De schildpad gaat terug naar zijn startpositie.
`?re 9Ø`	De schildpad draait naar rechts en is klaar om de middellijn van de straat te gaan tekenen.
`?herhaal 16 [pn vt 8 po vt 8]`	De schildpad tekent de stippellijn midden over de straat.
`?li 9Ø at 8`	De schildpad gaat naar het midden van de onderste weghelft.
`?re 9Ø`	De schildpad draait naar rechts.
`?zetvaart 1Ø`	De schildpad 'rijdt' weg.
`?zeg 1 ts`	Schildpad 1 luistert en komt tevoorschijn.
`?po vt 8`	Schildpad 1 gaat naar de bovenste weghelft.
`?li 9Ø`	De schildpad draait naar links.
`?zetvaart 1Ø`	De schildpad 'rijdt' weg.

Wil je graag dat de schildpadden eruit zien als auto's? Je hoeft ze dan alleen maar de vorm van een auto te geven.

Zelf een Vorm Bepalen voor de Schildpad

Je kunt ook zelf een vorm bedenken voor de schildpad. Hoe je de schildpad die vorm geeft, lees je hier. Je zult zien dat het heel gemakkelijk is, hoewel het soms een beetje ingewikkeld klinkt. Kijk eens naar vorm 1Ø:

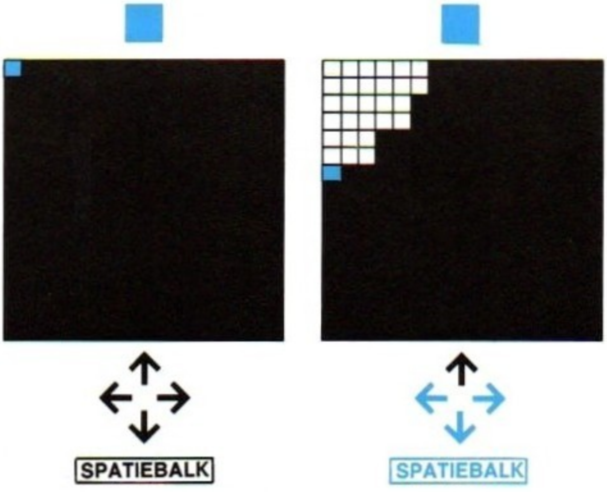
`?zeg allemaal ss sw`	Het scherm wordt schoongemaakt.
`?zeg Ø ts`	Een schildpad kiezen.
`?zetv 1Ø`	Een vorm kiezen voor de schildpad.
`?kleursp 1`	De schildpad een kleur geven. Denk eraan dat de schildpad nu zwart wordt als je een zwart/wit scherm gebruikt.
`?po vt 65 pn`	Zet de schildpad boven aan het scherm.

En dan nu iets nieuws:

`?ev 1Ø`	**ev** staat voor **e**dit **v**orm. Je kunt de schildpad hiermee zelf een vorm geven.

Op je beeldscherm moet nu een grote tekening staan van vorm 1Ø. De kleine, 'gewone' vorm van de schildpad staat ook nog op het scherm, precies boven deze Vorm-editor.

In de linker bovenhoek van de Vorm-editor staat een knipperend blokje, de *cursor*. Je verplaatst de cursor met de pijltoetsen. Druk nu op de **spatiebalk** en het vierkantje waarin de cursor staat, wordt wit (leeg). Druk nog een keer op de **spatiebalk** en het vierkantje wordt weer zwart. Verplaats de cursor door de Vorm-editor en druk opnieuw op de **spatiebalk**.

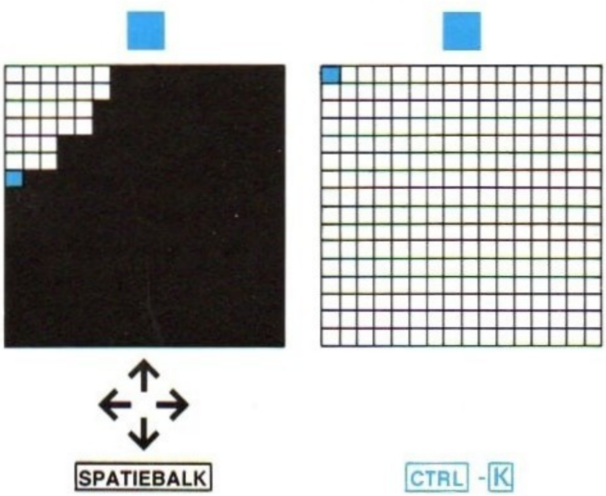


Dit is alles, meer hoef je niet te doen. Gemakkelijk of niet? Je ziet hoe de cursor tekent door vierkantjes te vullen en leeg te maken. Je kunt de vorm nu veranderen zoals je zelf wilt.

Ik vind deze vorm niet mooi. Kan ik er iets anders van maken? En deze Vorm-editor schoonmaken?

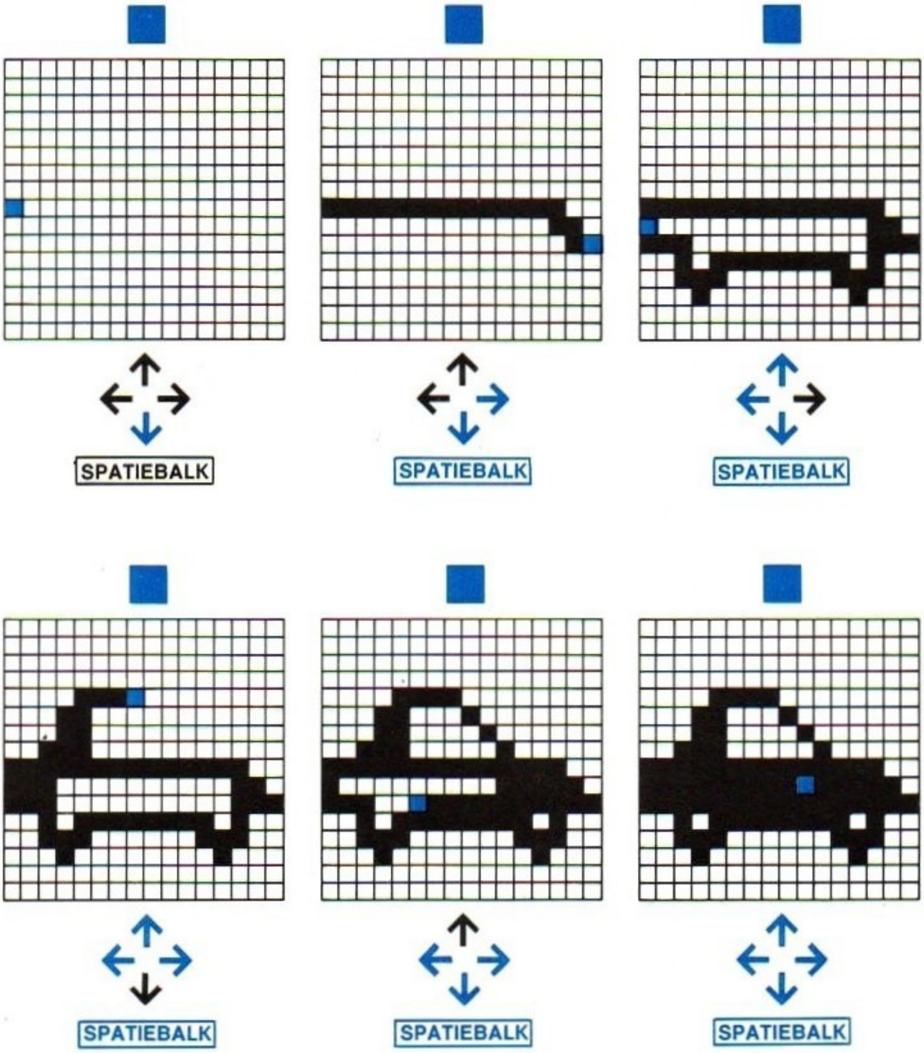
Je hoeft alleen maar de toetscombinatie CTRL-K *in te drukken. Zoals je ziet, bestaat de Vorm-editor uit een raster van 16 maal 16 vierkantjes.*

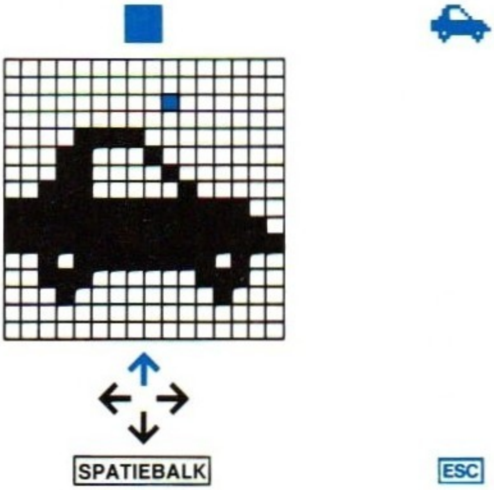
Een toetscombinatie wil zeggen dat twee toetsen gelijktijdig worden ingedrukt. Je houdt de CTRL *-toets ingedrukt en drukt dan op de toets* K *.*



Als je vorm 1Ø verandert en de veranderingen bevallen je niet, dan mag je opnieuw beginnen. Hiervoor gebruik je de toetscombinatie CTRL - STOP om de Vorm-editor te verlaten. Daarna typ je weer **ev 1Ø.**

Wanneer je klaar bent met je nieuwe vorm, druk je op de toets **ESC**.

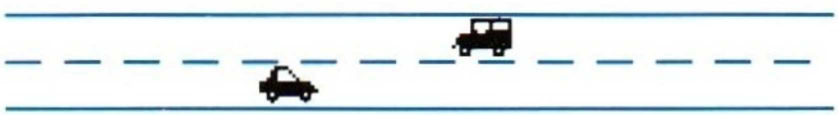




Als je op **ESC** drukt, vertel je de Vorm-editor dat je weg wilt gaan tot je een andere vorm wilt veranderen. Kijk naar de schildpad! Hij heeft nu de vorm 1Ø die je zojuist zelf hebt gemaakt.

Laten we nu verder gaan met je project. Maak de vorm van een auto en **zeg** twee schildpadden dat ze aan de opdracht **zetv 1Ø** moeten gehoorzamen. Daarna kun je ze over de straat laten rijden.

Misschien wil je nog een tweede vorm van een auto. Daarvoor moet je dan een andere vorm veranderen (bijvoorbeeld vorm 11).



Voordat je aan het volgende hoofdstuk begint, kun je gebruik maken van de opdracht **ms**.



Hoofdstuk 4

Procedures

Uiteindelijk werkt alles zoals je dat wil. Maar je moest wel een hele serie instructies verschillende keren typen. In dit hoofdstuk lees je hoe je dat extra werk kunt vermijden.

Logo een Nieuwe Procedure Leren

Stel dat je een hele reeks vierkanten wil tekenen. Dan zou het toch handig zijn, als je telkens alleen maar **vierkant** hoefde te typen. Laten we dat eens proberen. Misschien dat er iets gebeurt dat interessant is.

```
?vierkant
Ik ken geen vierkant
```

In plaats van een vierkant te tekenen, geeft Logo het bericht: **Ik ken geen vierkant.**

Goed, als Logo de opdracht **vierkant** niet kent, dan kun je hem dat *leren:*

Hiervoor gebruiken we de opdracht **leer**. Met leer vertel je Logo dat je zelf een procedure wilt maken. We spreken dan van een *proceduredefinitie.* Een proceduredefinitie is een serie instructies met een naam. Het woord **vierkant** zal de naam worden van onze proceduredefinitie.

Wanneer je op je scherm kijkt, zie je dit:

```
?leer vierkant
leer vierkant
eind
```

Je hebt de instructie **leer vierkant** getypt.

Logo zette deze twee regels op het scherm.

Het is jouw taak om tussen **leer vierkant** en **eind** de serie instructies in te vullen, die Logo vertellen wat jij hem met de opdracht **vierkant** wilt leren. Dit noemen we het *definiëren* van **vierkant.**

Druk op de **Return** -toets om een lege regel te maken en typ dan de volgende instructies:

```
leer vierkant
vt 50 re 90
vt 50 re 90
vt 50 re 90
vt 50 re 90
eind
```

Dit is de titelregel.

Typ dan de instructies die je Logo wilt leren.

Met **eind** vertel je Logo dat **vierkant** hier eindigt.

De schildpad blijft op zijn plaats staan omdat je tegen de Logo-editor praat, niet tegen de schildpad. Je weet dat, omdat er aan het begin van de regels geen vraagteken (?) meer staat.

> Als je een fout maakt, kun je die gemakkelijk verbeteren.
>
> Met de pijltoetsen verplaats je de cursor. Zet hem op het cijfer of de letter die fout is.
>
> Dan druk je op de **DEL** -toets om de fout weg te halen. Vervolgens kun je het juiste cijfer of de juiste letter typen.

Wanneer je klaar bent met het typen van de instructies, druk je op **ESC** om de Logo-editor te verlaten. Op het scherm verschijnt het bericht **vierkant gedefinieerd**. Hiermee geeft Logo aan dat hij nu de opdracht **vierkant** kent.

Probeer je nieuwe procedure nu maar uit:

```
?vierkant
```

Als alles goed is gegaan, moet er nu een vierkant op je scherm komen. Zo niet, lees dan hieronder verder. Je leert dan hoe je je proceduredefinitie moet veranderen.

Er gebeurt niets! Ik krijg het vraagteken niet terug op mijn scherm!

Druk op de **ESC** *-toets om de Logo-editor te verlaten en definieer de procedure* **vierkant.**

De Procedure Veranderen

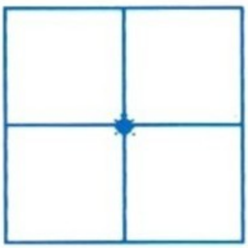
Wanneer je je procedure probeert, gebeurt er misschien iets wat je niet zo had bedoeld. Als je daarom iets wil veranderen, moet je opnieuw **leer vierkant** typen. De procedure **vierkant** verschijnt dan weer op je scherm in de Logo-editor.

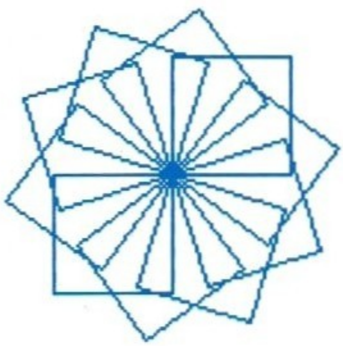
Nu kun je alle veranderingen maken die je wilt. Met de pijltoetsen verplaats je de cursor en met de DEL -toets en de BS -toets (BACK SPACE = spatie terug) kun je letters en cijfers op het scherm weghalen. Daarna typ je de juiste letters of cijfers.

Wanneer je klaar bent, druk je op ESC.

Probeer dit eens:

```
?herhaal ␣ [vierkant re ␣]
```



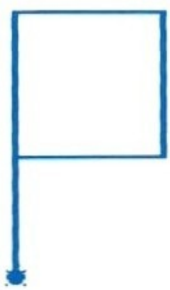


```
?herhaal 4 [vierkant re 9Ø]
```

```
?herhaal 1Ø [vierkant re 36]
```

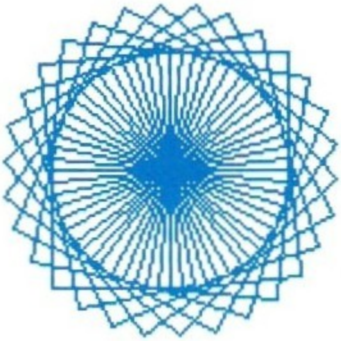
Je kunt de procedure **vierkant** ook binnen een andere proceduredefinitie gebruiken:

```
?leer vlag
leer vlag
at 4Ø vt 4Ø
vierkant at 4Ø
eind
```



```
?leer ster
leer ster
herhaal 30 [vierkant re 12]
eind
```

Kijk even naar deze lijst. Er staan twee instructies in, namelijk **vierkant** en **re 12.**



Je hoeft natuurlijk niet per se 30 en 12 als input te gebruiken. Je kunt ook je eigen ideeën proberen.

Vlag en **ster** werken alleen als je **vierkant** hebt gedefinieerd. In dit geval noemen we **vierkant** een *subprocedure* van **vlag** of **ster**. **Vlag** en **ster** noemen we dan de *superprocedures.*

Procedures met Inputs

Dit is de opdracht. → `vt` ← Dit is de input. `50`

```
vt 50
```

Samen noemen we ze een instructie.

Dit is ook een opdracht. ↓ ← Maar ze heeft geen input.

```
po
```

Het is een volledige instructie op zichzelf.

Bij sommige opdrachten moet je *altijd* een input geven. Bijvoorbeeld **vooruit** kan niet zonder input werken. De opdracht weet anders niet hoe ver de schildpad vooruit moet worden verplaatst. Maar een opdracht als **penop** vertelt de schildpad alleen maar dat hij zijn pen omhoog moet houden. Het is niet belangrijk hoe hoog de pen wordt opgetild, als hij maar geen strepen tekent op het scherm!

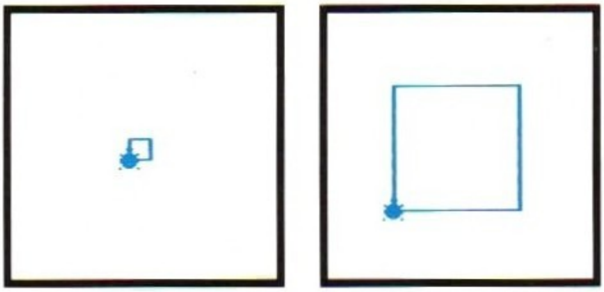
Je kunt iedere opdracht zien als een mannetje in de computer. Dit mannetje moet taken uitvoeren, en om dat te kunnen doen, heeft hij soms extra informatie nodig. Bijvoorbeeld het **vooruit**-mannetje moet weten hoe ver hij vooruit moet gaan.

Vierkant is een zelfde soort opdracht als **penop**: je hoeft niets anders te doen dan deze opdracht te typen. De schildpad tekent dan een vierkant waarvan elke zijde 5Ø stappen lang is.

Het zou mooi zijn als we een procedure **vierkant** hadden die als volgt werkte:

`?vierkant 1Ø` om een *klein* vierkantje te maken, en

`?vierkant 1ØØ` om een *grote* vierkant te maken.



Laten we eens beginnen met de oude **vierkant** weg te doen:

```
?weg "vierkant
```

Vergeet niet de dubbele aanhalingstekens te typen.

Controleer nu even of de vierkant inderdaad weg is:

```
?vierkant
Ik ken geen vierkant
```

Logo is **vierkant** vergeten - de procedure is weggehaald. Nu gaan we een nieuwe procedure **vierkant** definiëren. Start de Editor door **leer vierkant** te typen.

Oude procedure vierkant	Nieuwe procedure vierkant
leer vierkant	leer vierkant
vt 5Ø re 9Ø vt 5Ø re 9Ø vt 5Ø re 9Ø vt 5Ø re 9Ø eind	**Vt** ⎵ wat nu? Je kunt hier nu geen vaststaand getal invullen. Iedere keer dat je vierkant laat tekenen, wil je immers zelf kunnen bepalen hoe lang de zijden moeten worden. Daarom kun je een *naam* verzinnen voor de afmeting van het vierkant. Laten we dit maar gewoon **afmeting** noemen. Je typt de instructies nu als volgt in: vt :afmeting re 9Ø vt :afmeting re 9Ø vt :afmeting re 9Ø vt :afmeting re 9Ø eind Denk aan de dubbele punten.

Eén belangrijk ding: op de regel waarop we **leer** typen (dat is de titelregel) *in de Editor*, moeten we Logo vertellen dat deze procedure een input nodig heeft en dat **afmeting** de naam is die we voor de input gebruiken. Dat doe je zo:

Oude procedure vierkant	Nieuwe procedure vierkant
leer vierkant	leer vierkant :afmeting
vt 5Ø re 9Ø	vt :afmeting re 9Ø
vt 5Ø re 9Ø	vt :afmeting re 9Ø
vt 5Ø re 9Ø	vt :afmeting re 9Ø
vt 5Ø re 9Ø	vt :afmeting re 9Ø
eind	eind

Als je deze instructies hebt getypt, druk je op de **ESC** -toets om de Logo-editor te verlaten en je procedure te definiëren.

Probeer de nieuwe procedure **vierkant**:

```
?vierkant 10
?vierkant 20
?vierkant 30
?vierkant 40
?vierkant 50
?vierkant 60
.
.
.
```



Typ nu zelf wat getallen voor je vierkant.

Mijn procedures die de oude **vierkant** *gebruiken, werken nu niet!*

*Je moet die procedures (***vlag** *en* **ster***) veranderen en* **vierkant** *een input geven. Hieronder kun je zien hoe de procedure* **vlag** *veranderd moet worden:*

```
?leer vlag
at 40 vt 40
vierkant 50 at 40
eind
```

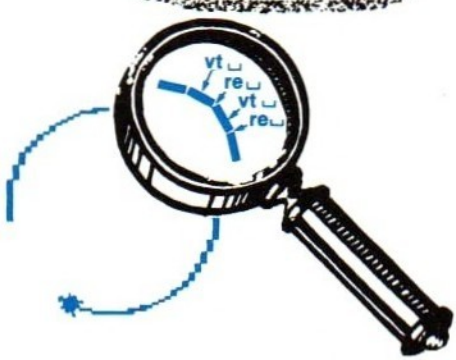
Cirkels en "Magische" Getallen

Hoe maak ik een cirkel?

Door schildpad te spelen. Je moet net doen alsof je een schildpad bent en in een rondje lopen.

Wat deed je?

Ik liep een klein stukje vooruit, draaide een heel klein beetje, liep weer een stukje vooruit, draaide weer een heel klein beetje, enzovoort. Ik heb dat heel vaak herhaald.




```
herhaal ␣ [vt 1 re 1]
```

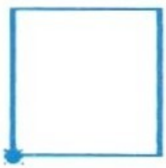
Maar hoe vaak moet de schildpad **[vt 1 re 1]** doen voor hij een cirkel heeft? In ieder geval heel vaak. Je kunt het zelf proberen en dan telkens kijken of de cirkel al rond is. Bijvoorbeeld:

```
?herhaal 100 [vt 1 re 1]
?herhaal 200 [vt 1 re 1]
```

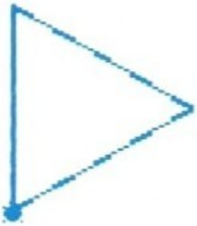


Je kunt verder gaan met gokken of ook iets anders proberen, zoals:

```
?herhaal 4 [vt ␣ re 90]
```



```
?herhaal 3 [vt ␣ re 120]
```



De schildpad voert de instructies **re 9Ø** vier keer uit om een vierkant te maken; **re 12Ø** voert hij drie keer uit om een driehoek te maken. Hierbij gaat de schildpad telkens 36Ø graden rond van beginpunt naar eindpunt, zodat het eindpunt eigenlijk hetzelfde is als het beginpunt. Dit is het *theorema van de totale schildpadtoer.*

Schildpadtaal	Mensentaal
herhaal 4 [vt 5Ø re 9Ø]	**4** keer **re 9Ø** is een draai van 36Ø graden: een vierkant.
herhaal 3 [vt 5Ø re 12Ø]	**3** keer **re 12Ø** is een draai van 36Ø graden: een driehoek.
Dus, een cirkel maken gaat zo:	
herhaal 36Ø [vt 1 re 1]	36Ø keer **re 1** is een draai van 36Ø graden: een cirkel.

De schildpad gaat 1 stap vooruit en draait dan 1 stap naar rechts. Dit herhaalt hij 36Ø keer, tot hij weer terug is op het punt waar hij startte. De cirkel is dan rond.

Dit idee gebruik je dus ook om "bochten" te maken. Het zijn dan gewoon onvolledige cirkels.

Subprocedures

Eerder in dit boek hebben we gewerkt met schildpadden die over de straat liepen. Dit is een ideaal project om in één of meer procedures te zetten. Het is heel gemakkelijk. Je verzint gewoon een naam voor de procedure en zet daar alle instructies in die je nodig hebt.

```
leer straat
ms
po vt 16
re 9Ø
pn vt 256
li 9Ø po
at 32 re 9Ø
pn vt 256
po thuis
re 9Ø
herhaal 16 [pn vt 8 po vt 8]
eind
```

Dit is de naam.

Deze instructies komen uit hoofdstuk 3, "Beweging, Vormen en Kleur".

Weet je het nog? Om de Logo-editor te verlaten, druk je op ESC. Daarmee definieer je de procedure.

OK, dat was dat, maar de procedure is een beetje moeilijk te volgen als je hem leest. Het is misschien een goed idee om sommige van de instructies in **straat** in een subprocedure te zetten.

```
leer straatkant
po vt 16 re 90
pn vt 256
po li 90 at 16
eind
```

In de Editor kun je meer dan één procedure definiëren. Maar je moet wel op de laatste regel van iedere procedure **eind** *typen.*

Nu moet je het volgende doen:

```
leer nieuwestraat
ms
straatkant
re 180
straatkant
eind
```

Druk op ESC om de procedures te definiëren.

Waarom ben je daar gestopt?

Ik wil even kijken hoe het eruit ziet. Ik maak het later af.

```
?nieuwestraat
```

Het werkt niet goed. Alleen de kanten van de straat worden getekend. Maar geen rijstroken en geen auto's. Een rare straat!

Maar de straatkanten werken goed. Laten we nu de rijstroken en de auto's maken.

Typ nu **leer nieuwestraat**. In de Editor verschijnt dan de procedure **nieuwestraat**.

```
?leer nieuwestraat
leer nieuwestraat
ms
straatkant
re 18Ø
straatkant
rijstroken  ←——  Nu werken we op deze regel.
auto's
eind
```

Hé, wat is dat? Dat zijn geen procedures. **Rijstroken**? **Auto's**?

Dat komt nog wel, let maar op!

Nu gaan we Logo leren wat **rijstroken** *zijn.*

```
leer rijstroken
po re 9Ø at 128
herhaal 16 [pn vt 8 po vt 8]
at 128 li 9Ø
eind

leer auto's
zeg Ø ts
po thuis
at 8 re 9Ø
zetvaart 1Ø
zeg 1 ts
po
vt 8 li 9Ø
zetvaart 1Ø
zeg [Ø 1]
eind
```

Druk op ESC om de Editor te verlaten.

Probeer deze procedures nu maar eens uit. Je typt dus **ss rijstroken** en daarna **ss auto's**. Daarna typ je nog een keer **ss** en vervolgens **nieuwestraat** om te controleren of de procedures ook samen werken.

Hier zijn nog twee nieuwe opdrachten

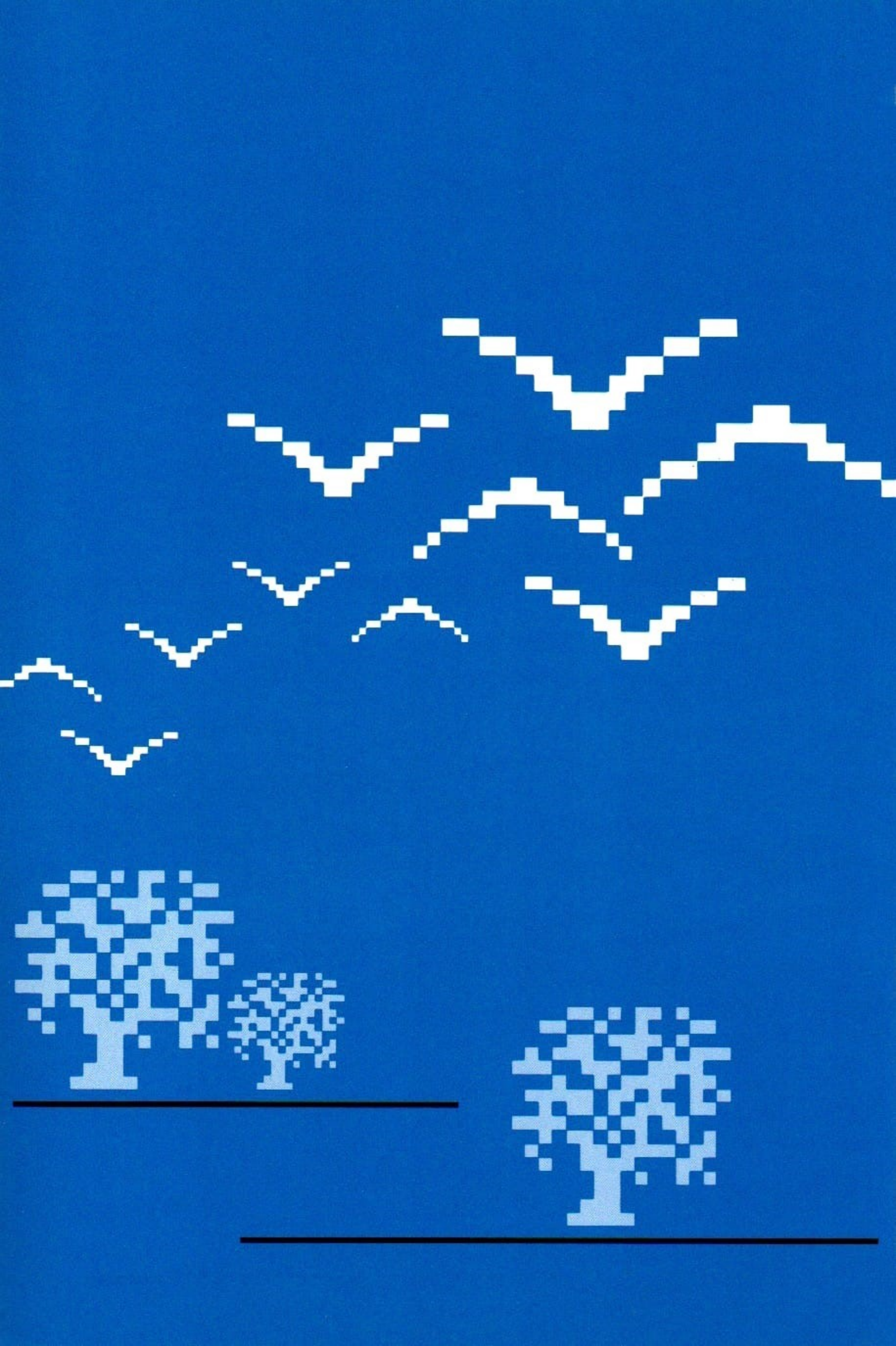
```
dats
dalles
```

Met de opdracht **dats** druk je de titelregels af van alle procedures die je tot nu toe hebt gedefinieerd. Met de opdracht **dalles** druk je de titels en de instructies af van alle proceduredefinities.

Afdrukken op Papier

Je proceduredefinities druk je als volgt af op je printer. *Probeer dit alleen wanneer er ook inderdaad een printer op je computer is aangesloten.*

?printer	Opent het printerkanaal.
?dalles	Zet al je proceduredefinities op het scherm en drukt ze af op de printer.
?printeruit	Sluit het printerkanaal.



Hoofdstuk 5

Reporters

Help! Ik veranderde een vorm met **zetv** *en een kleur met* **kleursp**, *maar ik weet niet meer wat ik heb veranderd.*

Maar je kunt toch wel zien welke vorm en kleur de schildpad op dit moment heeft.

Ik bedoel, dat ik niet meer weet welke getallen *ik heb gebruikt. Wat nu?*

Je kunt een nieuw soort primitieve procedure gebruiken, namelijk een zogeheten *reporter*. Zulke procedures noemen we zo omdat ze ergens verslag van doen. Je gebruikt ze als volgt:

```
?drukaf spkleur
```

← **Spkleur** is een reporter. Hij geeft de informatie die **drukaf** zichtbaar moet maken op het scherm.

↑ **Drukaf** is een opdracht die Logo vertelt iets op het scherm af te drukken, dus zichtbaar te maken.

Welke informatie is dat?

Probeer dit maar eens, dan zie je hoe het werkt:

```
?kleursp 3
?drukaf spkleur
3

?kleursp spkleur + 1
?drukaf spkleur
4

leer volgend
kleursp spkleur + 1
eind
```

Dat is slim: je telt 1 op bij het getal van **spkleur** *en stuurt het resultaat dan door naar* **kleursp**.

De opdracht **volgend** zal nu de kleur van de schildpad automatisch veranderen.

```
?volgend
?volgend
?volgend
```

Soms zie je dat de schildpad verdwijnt. Dat gebeurt wanneer hij dezelfde kleur aanneemt als de achtergrond op het scherm. Met andere woorden, de schildpad krijgt dan kleur Ø, waardoor hij onzichtbaar wordt. Als je dan nog een keer **volgend** typt, verschijnt hij weer.

```
?herhaal 1ØØ [volgend]
```

Nu gaat het zo snel, dat ik het niet kan volgen!

```
?herhaal 5Ø [volgend wacht 3Ø]
```

Een nieuwe opdracht. (→ wacht)

Oefen er maar even mee en geef **wacht** telkens een andere input. (→ 3Ø)

Even iets anders, probeer dit eens:

```
leer wow
volgend
wacht 30
wow
eind
```

Deze proceduredefinitie maakt gebruik van wat we *recursie* noemen. **Wow** vlak voor **eind** vertelt Logo om nog een keer **wow** te doen... en nog een keer... en nog een keer... en nog een keer...

Help! Hoe moet ik dit stoppen?

Gebruik de toets-combinatie CTRL - STOP .

Nog meer Reporters

Iedere opdracht die begint met **zet** heeft een reporter bij zich.

```
?zetvaart 55
?drukaf vaart
55
```

```
?zetv 6
?drukaf vormn
6
```

Je kunt met **vorm** dezelfde dingen doen die je hiervóór hebt gezien bij **kleur** (**spkleur** en **kleursp**).

```
?zetv 0
?herhaal 12 [zetv vormn + 1 w→
acht 100]
```

Wanneer je de schildpad weer zijn oorspronkelijke vorm wilt geven, typ je de instructie **zetv 36.**

Een van de belangrijkste reporters is **wie**, die vertelt welke schildpad op dit moment luistert.

```
?zeg 6
?drukaf wie
6

?zeg Ø
?drukaf wie
Ø
```

Kijk nu hoe je **wie** kunt gebruiken. Laten we eens tegen een hele serie schildpadden gaan praten:

`?zeg [Ø 1 2 3 4 5 6 7]`	
`?ts`	Ze komen allemaal te voorschijn.
`?kleursp 1`	Ze zijn allemaal zwart.
`?thuis`	Ze staan allemaal op één plek.
`?zetv 4`	Ze zijn allemaal vrachtauto's.

Zie je maar één vrachtauto in plaats van acht? Dat komt doordat ze allemaal boven elkaar staan op één plaats.

Als je ze alle acht wil zien, kun je ze bijvoorbeeld ieder een andere snelheid geven:

```
?zeg 1 zetvaart 1Ø
?zeg 2 zetvaart 2Ø
?zeg 3 zetvaart 3Ø
.
.
.
```

Maar je kunt je een hoop typewerk besparen door de opdrachten **elk** en **wie** te gebruiken:

```
?zeg [Ø 1 2 3 4 5 6 7]
?elk [zetvaart wie * 1Ø]
```

`elk`	`[zetvaart`	`wie`	`*`	`1Ø]`
Elke schildpad	krijgt een snelheid	van zijn eigen getal	maal	tien.

Hoe weet ik nu hoe snel iedere schildpad gaat? Ze gaan niet allemaal even snel.

Gebruik de reporter die bij **zetvaart** *hoort en de opdracht* **elk.**

```
?elk [drukaf vaart]
```

Een reporter geeft alleen informatie over de eerste schildpad van de groep schildpadden die luistert. Als je informatie wilt hebben over alle schildpadden die luisteren, gebruik je **elk**.

Als je de schildpadden wilt stoppen, gebruik je de instructie **zetvaart Ø**.

Als er maar een paar schildpadden stoppen, dan geef je de instructie **zeg allemaal**. *Daarna geef je de instructie* **zetvaart Ø.**

Voordat je aan het volgende hoofdstuk begint, kun je gebruik maken van de opdracht **ms.**

PRINT 2.5
-0.657894
(.3 *
48 *

Hoofdstuk 6

Logo als Rekenmachine

Probeer dit eens:

```
?drukaf 4 + 5
9
```

Logo heeft **4** en **5** bij elkaar opgeteld en de uitkomst **9** op het scherm gezet.

```
?re 18Ø - 135
```

Logo trekt **135** af van **18Ø** en gebruikt de uitkomst **45** als input bij **rechts**. Hoe ver draaide de schildpad naar rechts?

Logo heeft een ingebouwde rekenmachine!

Om te vermenigvuldigen gebruikt de computer het teken *. Dus als we 9Ø willen vermenigvuldigen met 4, typen we:

```
?drukaf 9Ø * 4
```

Om te delen gebruikt de computer het teken /. Dus als we 3Ø door 5 willen delen, typen we:

```
?drukaf 3Ø / 5
```

De rekenkundige *bewerkingen* zijn eigenlijk ook reporters. Eigenlijk zijn bewerkingen en reporters hetzelfde.

De uitkomsten van bewerkingen moeten ergens naartoe worden gestuurd. Waarheen, dat moet je Logo vertellen.

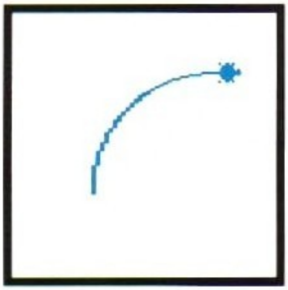
```
?5 + 2ØØ
Ik weet niet wat ik met 2Ø5 d→
oen moet
```

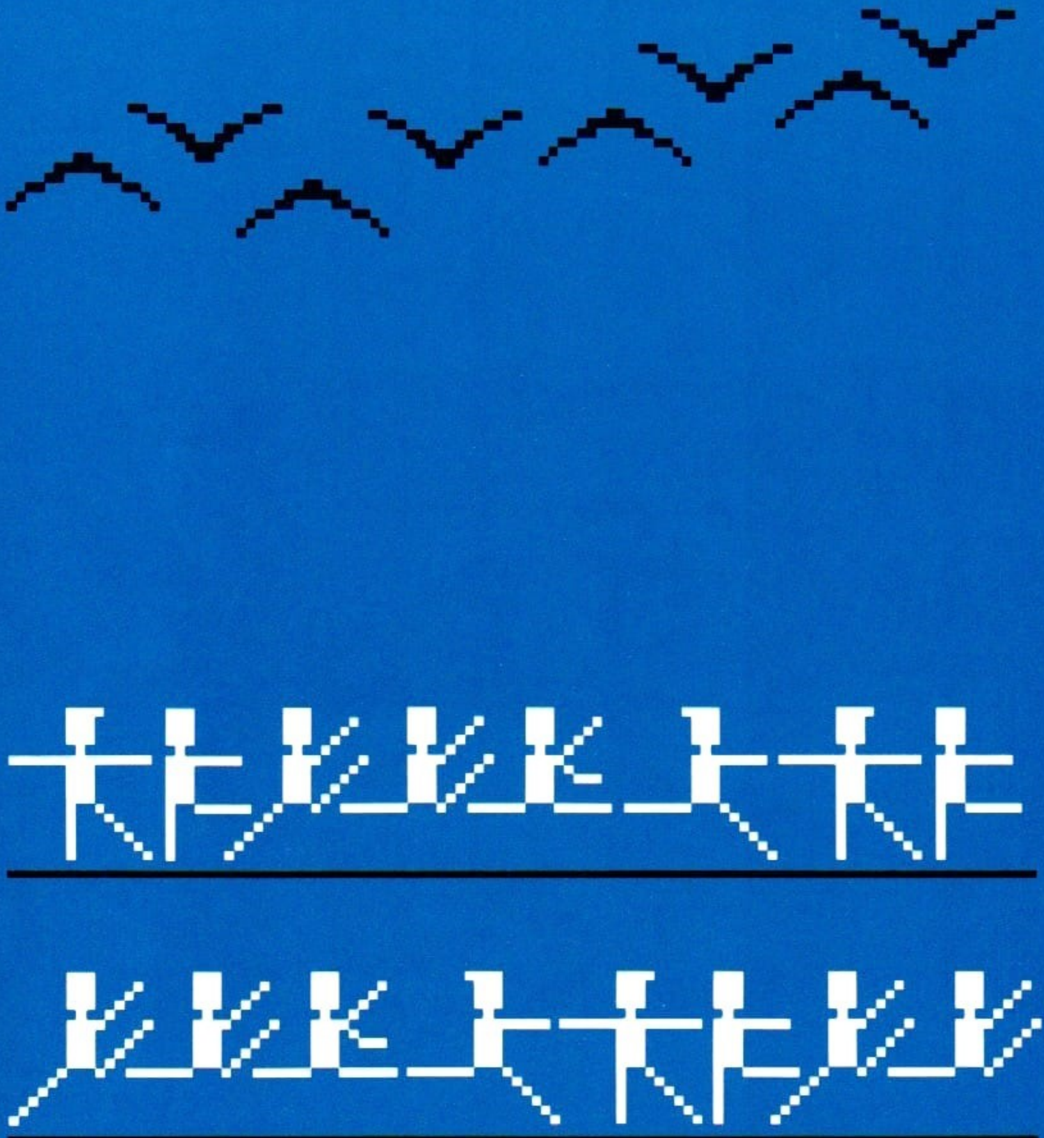
```
?drukaf 5 + 2ØØ
2Ø5
```

In plaats van zelf alsmaar te proberen hoe vaak je iets moet **herhalen**, kun je ook Logo het rekenwerk laten doen. Probeer dit maar eens:

```
?herhaal 36Ø / 4 [vt 1 re 1]
```

De schildpad tekent een kwart van een cirkel.



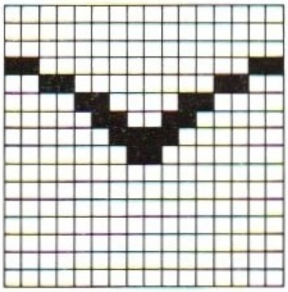


Hoofdstuk 7

Een
Film Maken

We gaan twee vormen maken en kiezen een nummer voor die vormen. Hier gebruiken we de nummers 14 en 15. Gebruik **ev** om vorm 14 te veranderen in deze vorm:

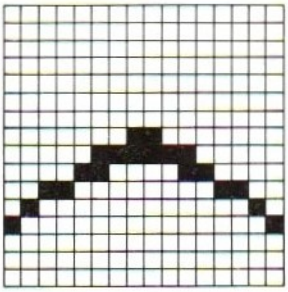
```
?ev 14
```



Tip: in de Vorm-editor kun je de oude vorm weghalen met de toetscombinatie CRTL-K.

Druk op ESC om de nieuwe vorm te bewaren.

Verander vorm 15 in deze vorm:



```
?zetv 14
?zetv 15
?zetv 14
?zetv 15
```

De schildpad gebruikt nu de twee vormen van vleugels en slaat ze op en neer als een vogel. Je kunt er als volgt een animatiefilmpje van maken:

```
leer vliegen
zetv 14
wacht 20
zetv 15
wacht 20
vliegen  ←——— Dit is weer die recursie.
eind
```

Toen je iedere instructie apart typte, regel voor regel en niet in een procedure, hoefde je niet de opdracht **wacht** te gebruiken. Waarom niet? Omdat je al automatisch wachtte tussen het typen van iedere instructie.

Controleer nu eerst of dit goed werkt:

```
?vliegen
```

Gebruik de toetscombinatie CTRL-STOP om de procedure te stoppen. Nu gaan we de vogel laten bewegen:

```
?penop
?zetvaart 20
?vliegen
```

Meer vogels maken:

Om meer vogels te maken, moet je eerst de procedure **vliegen** *stoppen met* CTRL-STOP.

```
?zeg 2 ts penop
?kleursp 1 thuis vt 50
?zeg [0 2]
?re 45 zetvaart 20 vliegen
```

Hoe krijg je nu alle vogels thuis?

Bedoel je midden op het scherm? Dat is gemakkelijk. Eerst stop je de procedure met CTRL-STOP *en dan typ je:* **zeg allemaal thuis**.

Geluidseffecten

Nu nog wat geluid terwijl de vogels vliegen.

Probeer dit maar:

```
?toon Ø 44Ø 15 15
```

Dit is de opdracht. (→ toon)

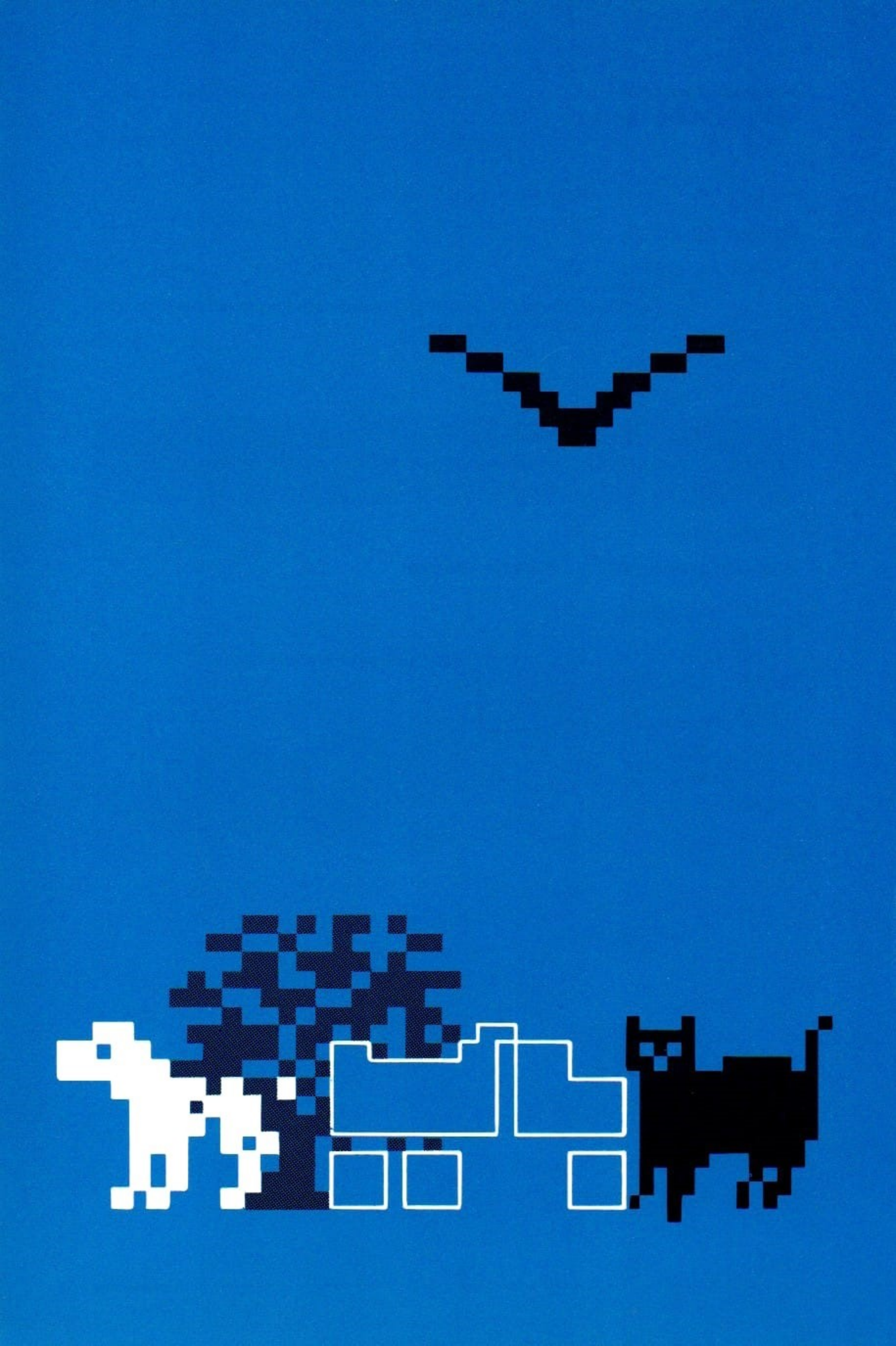
Dit zijn alle inputs voor **toon**. Het tweede getal geeft aan hoe hoog of hoe laag het geluid moet zijn. (→ Ø 44Ø 15 15)

> Hoor je niets? Als je een televisietoestel gebruikt, moet je het geluid harder draaien. Als je een monitor gebruikt, controleer dan of het geluidskanaal is aangesloten.

Voeg de geluidseffecten toe aan **vliegen:**

```
leer vliegen
zetv 14
toon Ø 44Ø 15 15
wacht 2Ø
zetv 15
toon Ø 88Ø 15 15
wacht 2Ø
vliegen
eind
```

Voeg deze twee regels toe. (→ de twee regels met toon)



Hoofdstuk 8

Namen Geven

Misschien ben je met de Vorm-editor bezig geweest om nieuwe vormen te maken of ben je aan het werk geweest met bestaande vormen. Elke vorm heeft zijn eigen nummer - 4 voor de vrachtauto, 1 voor het hartje of 8 voor de locomotief. Het kan best zijn dat je na een tijdje niet meer weet welk nummer bij welke vorm hoort. Het zou dus veel gemakkelijker zijn om de vormen een *naam* te geven in plaats van een nummer. Een naam vergeet je namelijk niet zomaar.

Gelukkig is dit met Logo mogelijk. Je gebruikt hiervoor de opdracht **noem**:

```
?noem 4 "vrachtauto
```

Let op de dubbele aanhalingstekens. Je zult dadelijk zien waarom ze er moeten staan.

Nu kun je zeggen:

```
?zetv :vrachtauto
```

Je zult dadelijk ook zien waarom je een dubbele punt moet typen.

Elke vorm kun je een naam geven:

```
?noem Ø "bal
?noem 3 "hond
```

Je kunt een naam kiezen **die je zelf wilt.**

Daarna kun je de vorm van de schildpad veranderen met:

```
?zetv :bal
?zetv :hond
```

Enzovoort.

In de computerwereld noemt men dit variabelen.

Overigens begrijpt Logo nog steeds de instructies:

```
?zetv Ø
?zetv 3
```

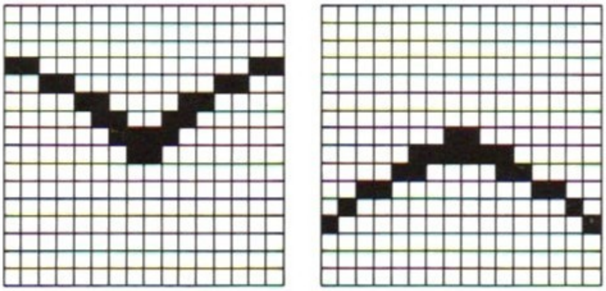
Je kunt dus ofwel *variabele namen* ofwel *getallen gebruiken.*

Namen voor Schildpadden en Namen voor Vormen

Je kunt nu het animatiefilmpje met de vogel veranderen, zodat het werkt met variabele namen in plaats van met getallen. Laat eerst één vogel vliegen. Kies een schildpad als vogel. Bijvoorbeeld schildpad Ø.

```
?noem Ø "vogel
```

Kies de twee vormnummers, 14 en 15, die je hebt gemaakt in Hoofdstuk 7, "Een Film Maken".



Geef ze nu een naam:

```
?noem 14 "omhoog
?noem 15 "omlaag
```

Variabele namen gebruiken	Nummers gebruiken
leer vliegen zetv :omhoog toon Ø 44Ø 15 15 wacht 2Ø zetv :omlaag toon Ø 88Ø 15 15 wacht 2Ø vliegen eind	leer vliegen zetv 14 toon Ø 44Ø 15 15 wacht 2Ø zetv 15 toon Ø 88Ø 15 15 wacht 2Ø vliegen eind
dan	dan
?zeg :vogel vliegen	?zeg Ø vliegen

Een Vlucht Vogels

Een naam geven is handig wanneer je een hele reeks schildpadden wilt gebruiken.

In dit project zullen tien vogels over het scherm vliegen. Schildpad Ø is de zon, schildpad 1 is de maan en de vogels zijn de schildpadden 2 tot en met 11. Het is nogal lastig als je steeds typen moet:

```
?zeg [2 3 4 5 6 7 8 9 1Ø 11]
```

dus geef je deze reeks schildpadden een naam:

```
?noem [2 3 4 5 6 7 8 9 1Ø 11]→
 "vogels
```

Nu kun je zeggen:

```
?zeg :vogels
```

Zo'n variabele naam is gemakkelijk te typen en gemakkelijk te onthouden. Je kunt ook het volgende proberen:

```
?noem Ø "zon
?noem 1 "maan
```

Bekijk nu dit eens:

```
leer vogels
zeg :vogels ts
po ss
kleursp 1
re 60
elk [vt wie * 15]
re 30
zetvaart 20
vliegen
eind
```

En de zon en de maan dan?

Waarom zet je die er niet zelf in?

Namen van Dingen

Waarschijnlijk heb je je inmiddels al afgevraagd waar de dubbele aanhalingstekens en de dubbele punten voor dienen. We zullen daar nu even op ingaan. Als je dit leest, mag je de computer even vergeten. Kijk maar even naar het volgende grapje:

Jij zegt tegen iemand:	Zeg je naam.
Stel dat die persoon dan zegt:	Barbara.
Dan zeg jij:	Ik zei niet **Zeg Barbara,** maar ik zei **Zeg je naam.**
Misschien zegt Barbara dan:	Zeg je naam.
Dan zeg jij:	Oh, heet jij zo? **Zeg je naam.**

Er bestaan verschillende vormen van dit grapje. Bijvoorbeeld:

Jij:	Wat is de langste rivier van Nederland?
Slachtoffer:	De Rijn.
Jij:	Spel het eens.
Slachtoffer:	R-I-J-N.
Jij:	O ja? Ik dacht dat het werd gespeld als H-E-T.

"Spel het eens" kan hier twee dingen betekenen:
spel het *woord* het: H-E-T

of

spel het ding *waar het woord het voor staat.*

Mensen met wie je praat, begrijpen meestal uit de klank van je stem wat je bedoelt. Als je het opschrijft, kan dat niet meer, maar dan kun je leestekens gebruiken om de betekenis duidelijk te maken. Bijvoorbeeld:

Spel het	betekent: spel de naam van het ding waar je het over hebt.
Spel "het"	betekent: spel het woordje "het" zelf.

Logo heeft zulke leestekens ook nodig. Voor Logo gebruik je dubbele punten en dubbele aanhalingstekens.

De aanhalingstekens gebruik je om aan te geven dat je het woord zelf bedoelt.

Een dubbele punt gebruik je om aan te geven dat je het ding bedoelt waar het woord voor staat.

```
?noem 55 "leeftijd

?drukaf "leeftijd
leeftijd
```

Logo zet het woord **leeftijd** op het scherm.

```
?drukaf :leeftijd
55
```

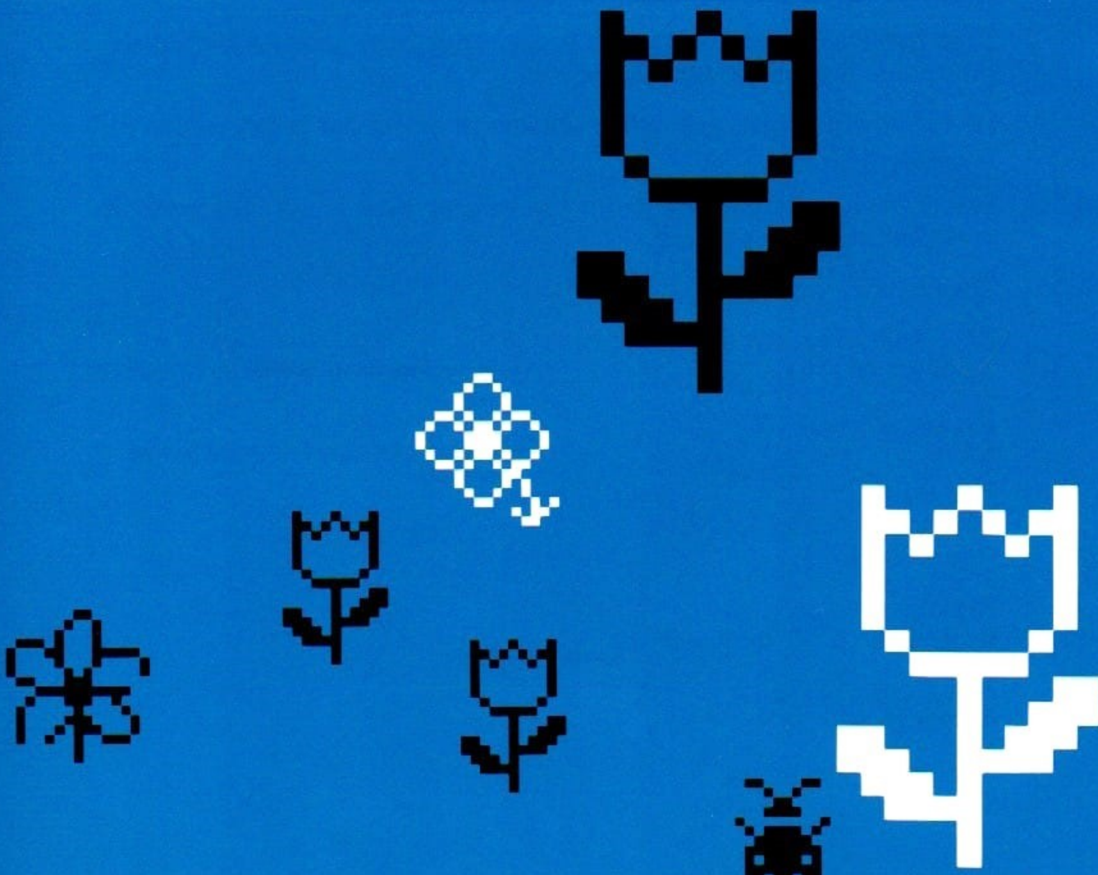
Logo laat zien waar het woord **leeftijd voor staat.**

```
?drukaf "vogels
vogels

?drukaf :vogels
2 3 4 5 6 7 8 9 10 11

?drukaf "omhoog
omhoog

?drukaf :omhoog
14
```



Hoofdstuk 9

Geheugen

Mensen hebben twee manieren om dingen te onthouden. We kunnen ze in "ons hoofd" houden of we kunnen ze opschrijven. Logo heeft ook twee manieren om dingen te onthouden.

Logo heeft een geheugenonderdeel dat we het "werkgeheugen" noemen. Als je een procedure definieert, iets een naam geeft of een vorm verandert, dan zet Logo dit in het werkgeheugen. Zolang de computer aan staat, herinnert Logo zich alles wat in het werkgeheugen staat. Daarna niet meer.

Als je nu wil dat Logo zich iets blijft herinneren, dus ook nadat je de computer hebt uit gezet, dan moet je het opschrijven op een manier die Logo begrijpt. Dit doe je door een kopie van een procedure, een variabele of een tekening als een pakketje op te slaan op een schijf of een cassetteband. Zo'n pakketje noemen we een file en deze handeling noemen we het *bewaren* van je werk. Een schijf of een cassetteband kun je zien als een notitieboekje van de computer.

Opmerking: als je geen schijfeenheid of een data recorder hebt, kun je je werk niet bewaren. Logo vergeet alles wanneer je de computer uit zet. De schijfeenheid of de data recorder moet op de computer worden aangesloten *voordat* je de computer aan zet.

```
?bewaar "vandaag
```

Als je een data recorder gebruikt, moet je **cbewaar** typen in plaats van **bewaar.**

De opdracht **bewaar** (of **cbewaar** als je een data recorder gebruikt) kopieert alle procedures en variabelen die in het werkgeheugen zitten naar de schijf (of de cassetteband) en noemt ze **vandaag** (je kunt een naam kiezen die je zelf wilt).

Als je je werk bewaart op een cassetteband moet je aan de volgende dingen denken:

Schrijf voor jezelf de naam van de file op en de plaats van de file op de cassetteband.

Druk eerst op de data recorder de knoppen **RECORD** en **PLAY** in *voordat* je **cbewaar** gebruikt.

Als je nu je computer uit zet, zul je je werk niet kwijt zijn. Het is op een schijf of cassetteband vastgelegd in een file die de naam **vandaag** heeft.

Wanneer je de computer weer aan zet, typ je:

```
?laad "vandaag
```

Als je een data recorder gebruikt, moet je **claad** typen in plaats van **laad.**

Logo kopieert alle procedures en variabelen van de file **vandaag** in het werkgeheugen.

Als je je werk laadt vanaf een cassetteband, moet je aan de volgende dingen denken:

Zet de cassetteband op een plaats *vóór* de file.

Druk op de data recorder de knop **PLAY** in.

Als je niet alles wilt bewaren dat in het werkgeheugen staat, moet je de rest weghalen. Gebruik **dats (voor d**ruk**a**f **t**itel**s**) om de titels van de proceduredefinities te zien en **dann (d**ruk**a**f **n**ame**n**) om de variabele namen en waarden te zien. Bepaal welke procedures of variabele namen je niet wilt hebben (bijvoorbeeld de procedure **vierkant** en de variabele **vrachtauto**) en typ dan:

```
?weg "vierkant
?nweg "vrachtauto
```

nweg staat voor variabele **n**aam **weg.**

Daarna kun je de inhoud van het werkgeheugen bewaren.

Misschien vergeet ik hoe ik mijn file heb genoemd.

Probeer **inhoud** *eens. Deze opdracht zet de namen op het scherm van alle files die op de schijf staan. De inhoud van de files zie je niet; je ziet alleen de namen.*

Inhoud *werkt alleen als je een schijfeenheid hebt. Als je een data recorder hebt, zul je zelf moeten bijhouden welke files je op de band hebt bewaard en waar ze precies staan.*

Plaatjes Bewaren

Als je een schijfeenheid op je computer hebt aangesloten, kun je ook de tekeningen of plaatjes bewaren die de schildpad op het scherm maakt.

Opmerking: tekeningen kun je alleen bewaren op een schijf, niet op een cassetteband.

De opdracht **bewaarpl** (**bewaar pl**aatje) kopieert alles wat op het scherm staat (inclusief tekst) naar een file op de schijf. Misschien wil je de volgende opdracht gebruiken:

```
?tw
```

tw staat voor tekst weg.

Deze opdracht haalt alle tekst van het scherm, maar laat alle tekeningen staan. Je kunt **tw** vóór **bewaarpl** zetten op dezelfde regel, zodat je geen tekst op je scherm hebt staan wanneer je het plaatje bewaart.

```
?tw bewaarpl "tekening
```

Of een andere naam, die je zelf wilt.

Wanneer je de computer weer aan zet of als je de tekening weer op je scherm wilt halen, typ je het volgende:

```
?laadpl "tekening
```

Nieuwe Vormen van de Schildpad Bewaren

Het bewaren van nieuwe vormen van de schildpad is iets ingewikkelder.

Logo gebruikt getallen om het patroon te onthouden dat de vorm bepaalt. Deze getallen werken als een code. Wanneer je nu een nieuwe vorm wilt bewaren, dan moet je de getallen bewaren. Dit doe je met de opdracht **pakv** (van **pak v**orm), die de getallen voor een vormnummer pakt.

Je weet waarschijnlijk nog wel dat we bij de procedure **vliegen** de vormen 14 en 15 als de vleugels gebruikten.

```
?noem pakv 14 "vleugelsop
?noem pakv 15 "vleugelsaf
```

De getallen voor iedere vorm hebben nu een naam. De volgende stap is het bewaren van alle namen en wat bij die namen hoort. Voor het bewaren van de namen gebruik je de opdracht **bewaar** (of **cbewaar** als je een data recorder hebt). Denk eraan dat alle procedures in het werkgeheugen ook worden bewaard in dezelfde file, tenzij je ze eerst weghaalt.

```
?bewaar "vleugelvormen
```

Als je nu de computer uit zet, zullen de nieuwe vormen niet verloren gaan. Ze worden bewaard op schijf of cassetteband in een file met de naam **vleugelvormen.**

Wanneer je de computer weer aan zet, typ je:

```
?laad "vleugelvormen
```

en

```
?neemv 14 :vleugelsop
?neemv 15 :vleugelsaf
```

Neemv (van **neem v**orm) vertelt vorm 14 dat ze het patroon moet aannemen dat we **vleugelsop** hebben genoemd. En vorm 15 krijgt het patroon dat we **vleugelsaf** hebben genoemd.

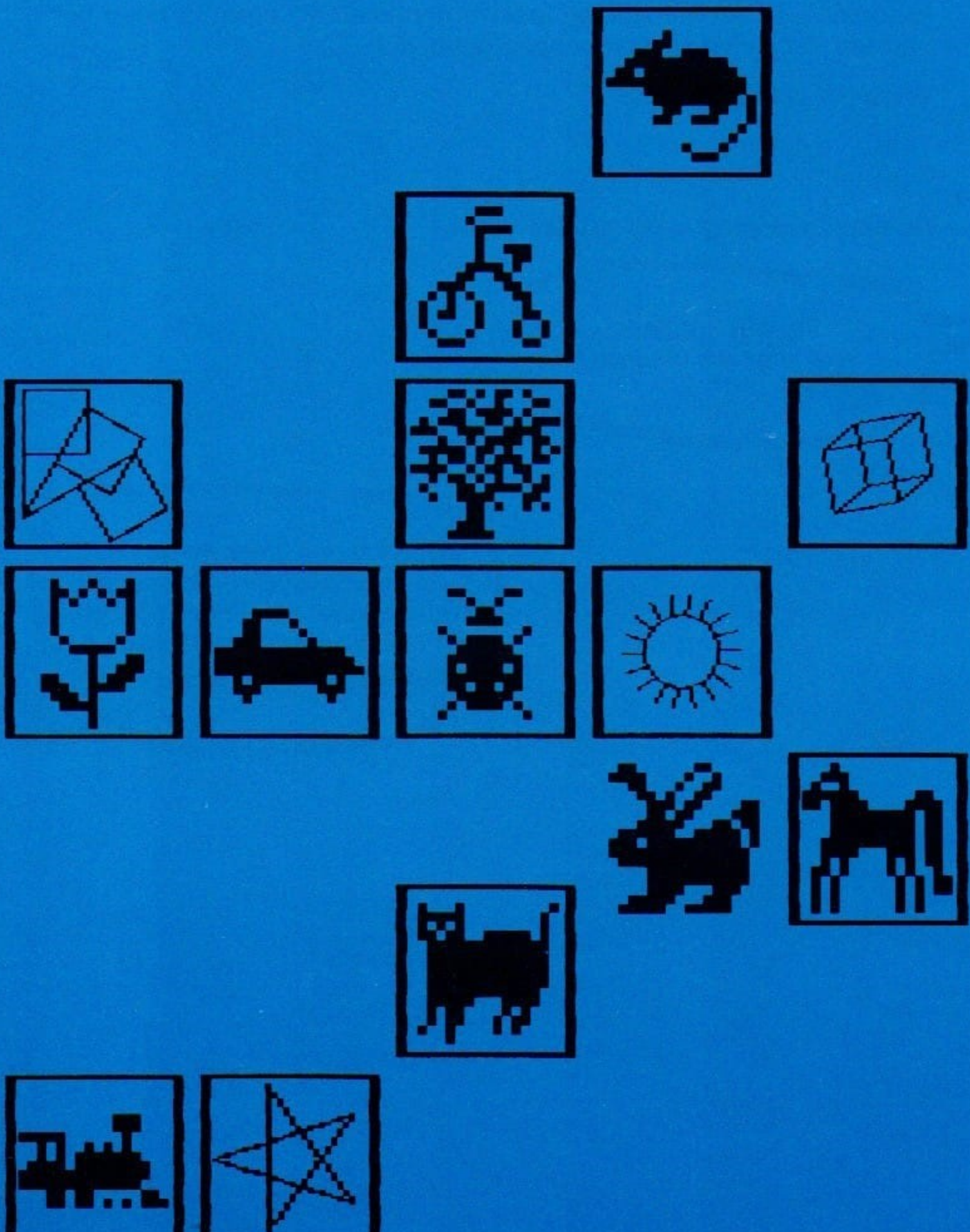
Wanneer je typt:

```
?zetv 14
?zetv 15
```

De schildpad vliegt weer als een vogel.

Zo, was het dat?
Wat gaan we nu doen?

Nee, dit is nog maar het begin. Nu is het jouw beurt. Gebruik je fantasie maar en alles wat je denkt dat je kunt gebruiken. (Vergeet ook niet het Beknopt Naslagwerk.*)*



Hoofdstuk 10

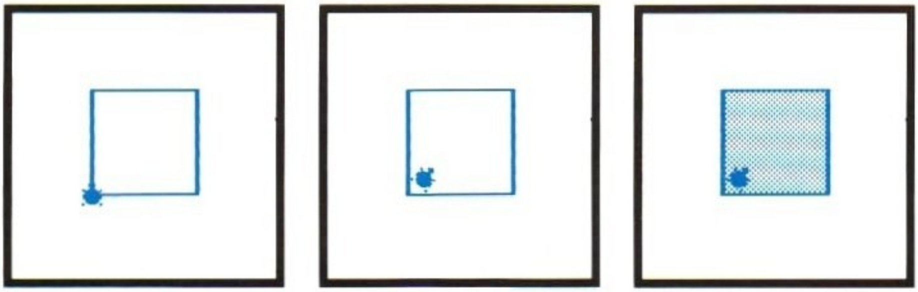
P.S.
Stof om mee te Experimenteren

Hieronder vind je een paar projecten waar je een beetje mee kunt experimenteren. Ze bevatten een paar nieuwe opdrachten en wat nieuwe ideeën. Je kunt er mee werken en zult spelenderwijs ontdekken wat je er allemaal mee kunt doen.

Een Vierkant Invullen

Probeer het volgende eens en gebruik daarbij de procedure **vierkant** die je in Hoofdstuk 4 hebt gedefinieerd:

```
?vierkant 5Ø
?po re 45 vt 5
?pn zetpk 7 vul
?zetpk 14 vul
```



Een andere manier om de schildpadden te stoppen en te starten

Probeer het volgende maar:

```
?zeg allemaal
?ss ts po
?elk [richt 12 * wie]
?zetvaart 5Ø
?halt
?start
```

Je kunt de schildpad ook een paar instructies geven - zoals **re 45** of **vt 3Ø** - nadat je **halt** hebt gebruikt en voordat je **start** gebruikt.

Botsende Schildpadden

Hier vind je een manier om de schildpadden te laten botsen. Je begint met alle afbeeldingen terug te zetten in de beginstand:

```
?ms
```

Nu luistert alleen nog schildpad Ø. Zeg een andere schildpad dat hij ook moet luisteren, te voorschijn moet komen, zijn pen omhoog moet houden en moet gaan lopen:

```
?zeg 1 ts po zetvaart 3Ø
```

Telkens wanneer hij over het scherm loopt, zal deze schildpad over schildpad Ø heen lopen.

Nu iets nieuws:

```
?raak.aan Ø 1 [re 18Ø]
```

De opdracht **raak.aan** maakt een *demoon* wakker die erop let of twee schildpadden tegen elkaar botsen, in dit geval schildpad Ø en 1. Zodra ze elkaar raken, geeft de demoon de instructielijst door aan de schildpad(den).

Als je de demoon weer wilt weghalen, typ je:

```
?demweg
```

demweg staat voor **dem**onen **weg.**

Probeer nu de volgende procedure eens. Je gebruikt hiermee twee demonen:

```
leer botsen
zeg Ø
ts po
vt 6Ø
zeg 1
ts po
at 6Ø
zeg 2
ts po
raak.aan 2 1 [re 18Ø]
raak.aan 2 Ø [re 18Ø]
zetvaart 2Ø
eind
```

Door deze twee demonen draait schildpad 2 **rechts 18Ø** zodra hij schildpad 1 of schildpad Ø raakt.

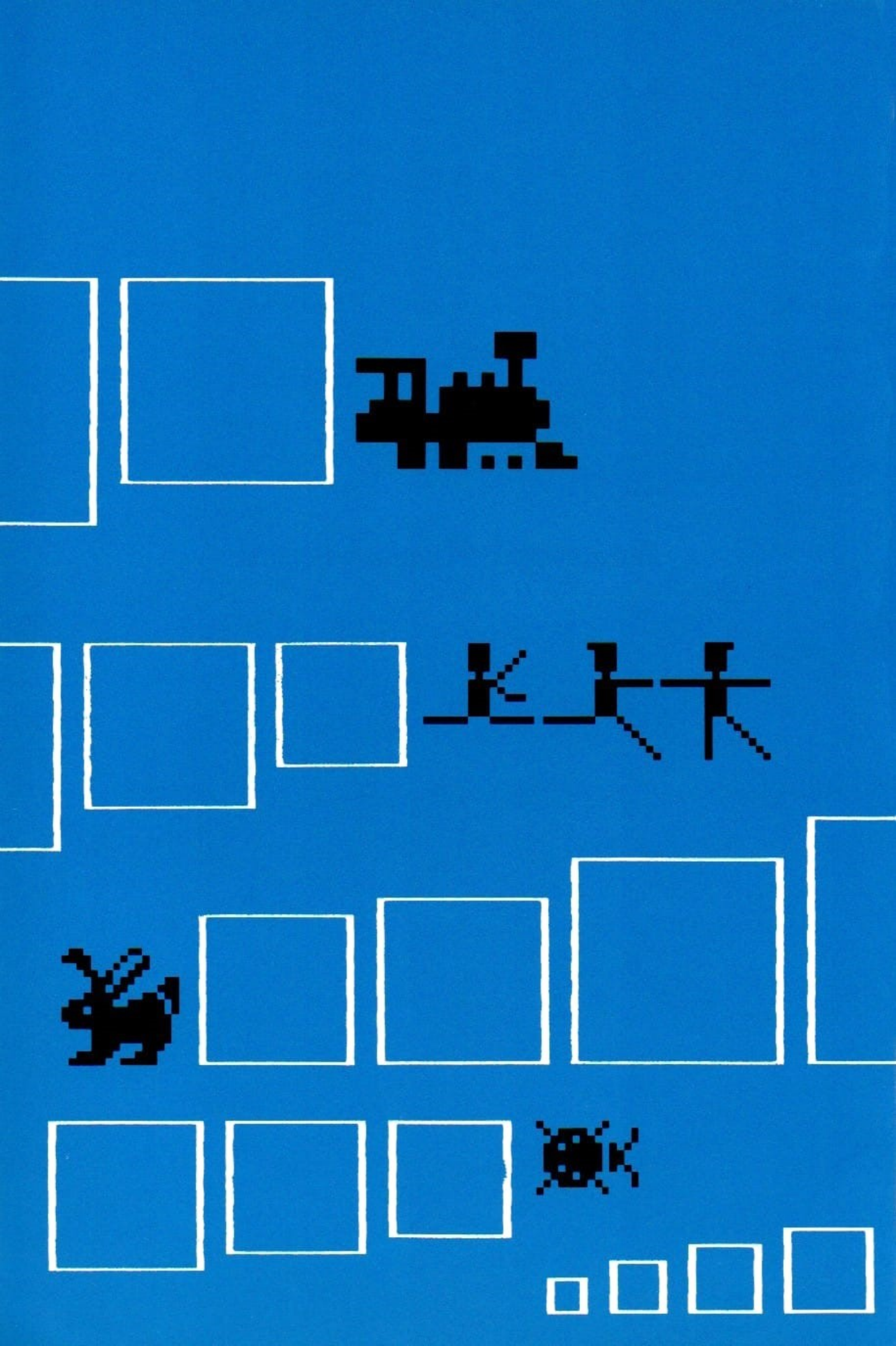
De opdracht **raak.aan** heeft drie inputs nodig: twee nummers voor schildpadden en een lijst met instructies die moeten worden uitgevoerd wanneer de schildpadden elkaar raken.

De opdracht **demweg** haalt alle demonen weg. De opdracht **ms** haalt ook alle demonen weg en zet de afbeeldingen op het scherm terug in hun beginstand.

Doe dit laatste nu en probeer dan deze procedure.

```
?ms
?botsen
```

Beknopt Naslagwerk



Inleiding

Het Beknopte Naslagwerk bestaat uit de volgende gedeelten:

Kleur en Vorm in MSX-Logo; hierin worden de verschillende kleuren, schildpadvormen en demonen beschreven die je in MSX-Logo kunt gebruiken.

Toetsen voor het Editen van Tekst en Vormen; hierin wordt beschreven welke toetsen je gebruikt in de Logo-editor en de Vorm-editor.

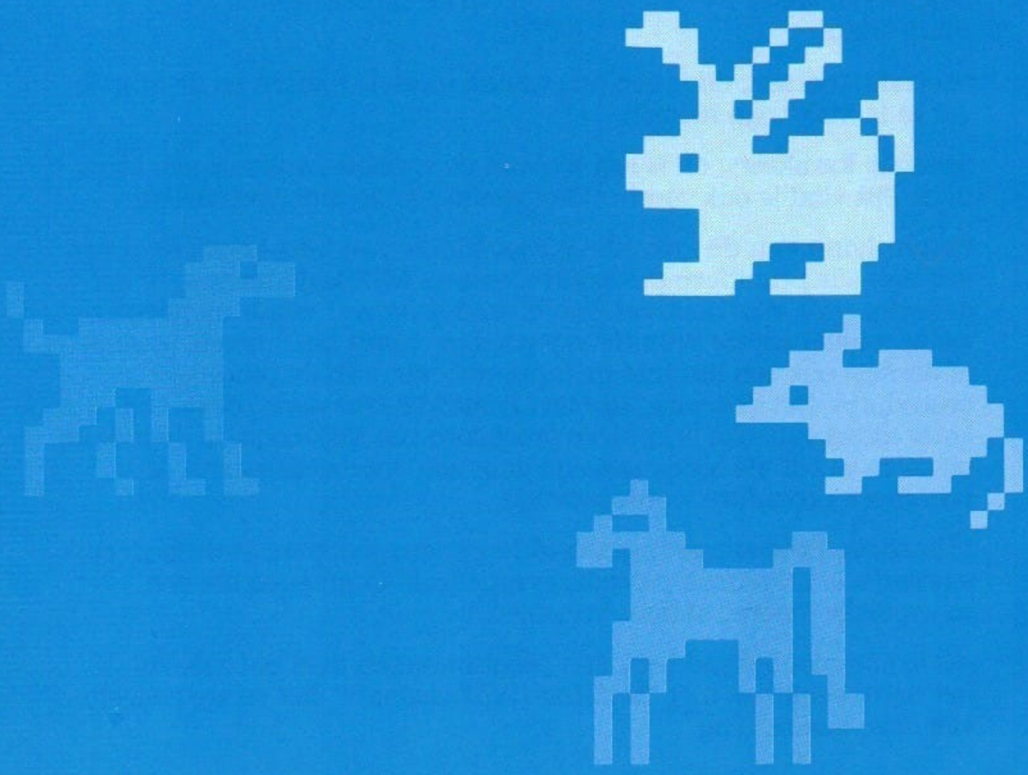
Speciale Toetsen; dit is een lijst van de speciale toetsen en de functie die ze in MSX-Logo hebben.

Speciale Karakters; dit is een lijst van de speciale karakters van Logo. In de lijst vind je ook voorbeelden van hoe je ze moet gebruiken.

Beschrijving van de Primitieve Procedures van MSX-Logo; dit is een lijst met alle primitieve procedures waaruit MSX-Logo bestaat. In het gedeelte "*Hoe We Primitieve Procedures Beschrijven*" wordt uitgelegd op welke manier iedere primitieve procedure wordt beschreven. Je vindt in dit deel ook een lijst met de input-woorden van de primitieve procedures. Het gedeelte "*Lijst met Primitieve Procedures*" geeft een korte uitleg van elke primitieve procedure van MSX-Logo. In de meeste gevallen wordt die uitleg gevolgd door een voorbeeld waarin de primitieve procedure wordt gebruikt.

Primitieve Procedures Ingedeeld in Functiegroepen; in dit gedeelte worden alle primitieve procedures ingedeeld in groepen die een soortgelijke functie, of taak, vervullen.

Als je nog niet bekend bent met programmeren in MSX-Logo, kun je het beste beginnen met "*MSX-Logo Schildpadtaal*", het eerste gedeelte van deze handleiding.



Kleur en Vorm in MSX-Logo

Kleuren

MSX-Logo heeft 16 kleuren, die je kunt gebruiken als kleur voor de schildpadden, voor de pennen waarmee ze tekenen, voor de achtergrond en voor de tekst. (Zie de tabel met kleuren hieronder.)

Als je begint met Logo, zijn de schildpadden en hun pennen wit (kleur nummer 15) en de achtergrond van het scherm is blauw (kleur nummer 5). Je kunt prachtige effecten bereiken als je de kleuren van de achtergrond, van de schildpadden en van hun pennen verandert.

De kleuren van de schildpadden kun je veranderen met **kleursp** (van **kleur s**childp**ad**), de kleur van hun pennen met **zetpk** (van **zet p**en**k**leur). Met **kleurscherm** kun je de **kleur** van de achtergrond (het **scherm)** veranderen en met **verkleur** de kleur van ieder gewenst plaatje of tekst op het scherm.

Opmerking: als de schildpad of een pen doorschijnend is (kleur Ø), of dezelfde kleur heeft als de achtergrond, lijkt hij onzichtbaar.

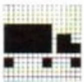
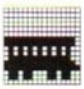
Kleurentabel

Nummer	Kleur	Nummer	Kleur
Ø	Doorschijnend	8	Middelrood
1	Zwart	9	Lichtrood
2	Middelgroen	1Ø	Geel
3	Lichtgroen	11	Lichtgeel
4	Donkerblauw	12	Donkergroen
5	Lichtblauw	13	Magenta (Paars)
6	Donkerrood	14	Grijs
7	Cyaan (Felblauw)	15	Wit

Schildpadvormen

Als je met Logo begint, hebben alle schildpadden de standaardvorm van een schildpad. Je kunt de schildpad bij MSX-Logo echter 6Ø verschillende vormen geven, genummerd van Ø tot 59. (Zie de tabel met schildpadvormen hieronder.) Met de opdracht **zetv** (van **zet v**orm) kun je de schildpad een bepaalde vorm geven. Je kunt alle 6Ø vormen veranderen met de Vorm-editor.

Vormentabel

Nummer	Vorm	
Ø		Bal
1		Hart
2		Kat
3		Hond
4		Vrachtauto
5		Raket
6		Stenen muur
7		Helikopter
8		Locomotief
9		Treinwagon
1Ø-35		Blok
36-59		Schildpadvorm in verschillende richtingen

Demonen

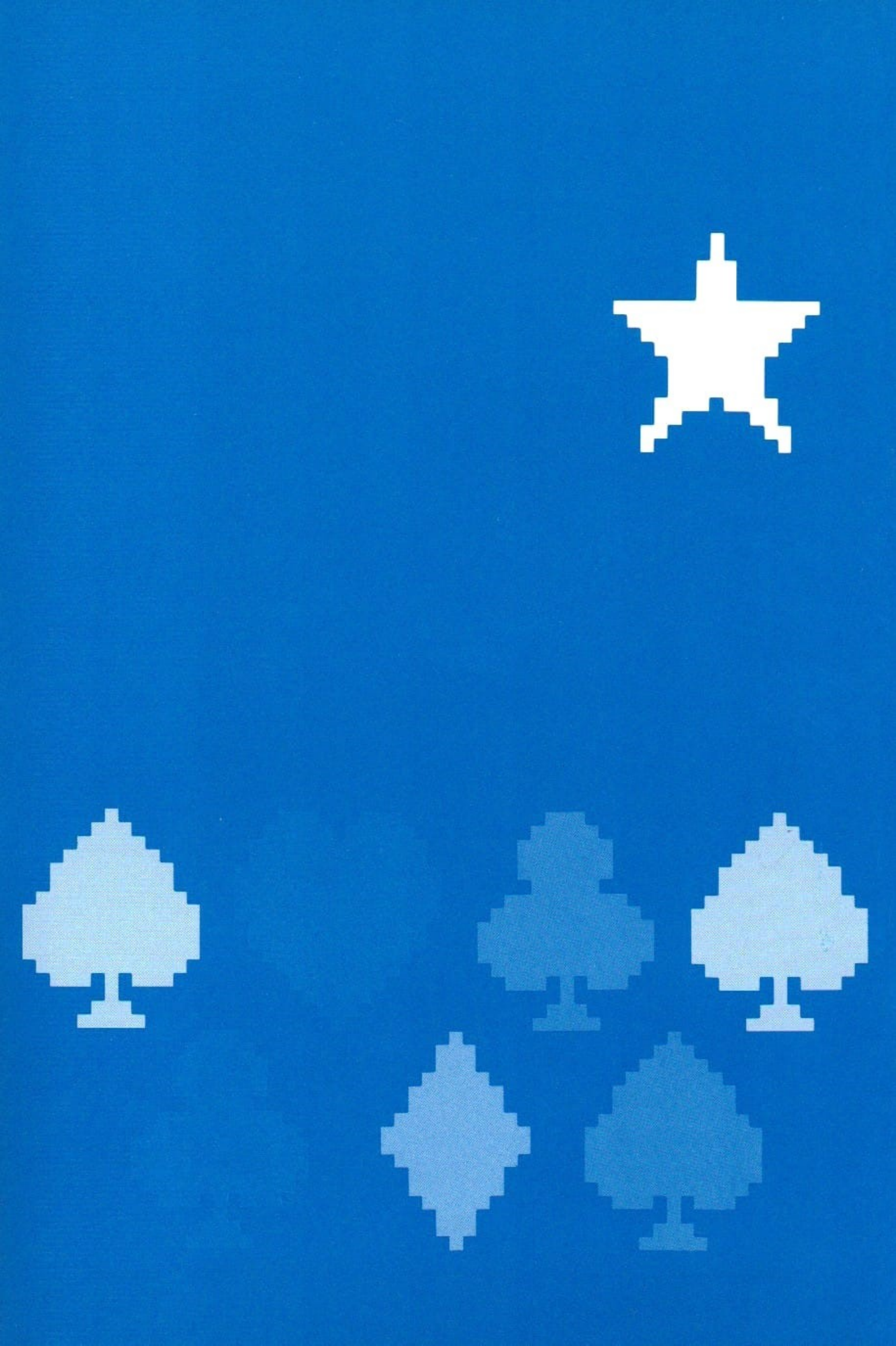
Een demoon is een bijzonder soort hulpmiddel bij het programmeren. Een demoon werkt los van alle programma's. Hij wacht tot er iets gebeurt of tot er een botsing is tussen twee schildpadden. Als er iets gebeurt, of als twee schildpadden botsen, dan onderbreekt de demoon de procedures die op dat moment worden uitgevoerd en voert hij zelf zijn instructielijst uit. Als de demoon klaar is met zijn instructies, gaat het vorige programma verder waar het gebleven was.

De twee primitieve procedures waarmee je demonen kunt maken, zijn **raak.aan** en **wanneer**. **Raak.aan**-demonen wachten op een botsing tussen twee schildpadden. **Wanneer**-demonen wachten tot er iets bepaalds gebeurt. Hieronder vind je een tabel, waarin je kunt zien op welke gebeurtenissen een **wanneer**-demoon kan wachten.

Een demoon blijft wachten op een gebeurtenis of een botsing totdat je hem wegdoet. Je kunt demonen wegdoen met de opdracht **demweg**, of door een lege instructielijst te geven als input voor **raak.aan** of **wanneer**.

Gebeurtenissentabel

Nummer	Gebeurtenis
Ø	Ongeveer één keer per seconde
1	Er wordt op een toets gedrukt
2	Er wordt op een knop op joystick 1 gedrukt
3	Er wordt op een knop op joystick 2 gedrukt
4	Joystick 1 wordt bewogen
5	Joystick 2 wordt bewogen



Toetsen voor het Editen van Tekst en Vormen

De Logo-editor

Je gebruikt de Logo-editor voor het definiëren en editen van procedures en variabelen, die daarna in het werkgeheugen worden ingevoerd. Je start de editor met de opdrachten **leer**, **edit** en **ednn** (van **ed**it **n**ame**n** van variabelen).

Als je in de editor bent, kun je de volgende edit-toetsen gebruiken:

Editen

• →	Verplaatst de cursor een positie naar rechts.
• ←	Verplaatst de cursor een positie naar links.
• ↓	Verplaatst de cursor een regel naar beneden.
• ↑	Verplaatst de cursor een regel naar boven.
• CTRL-A	Verplaatst de cursor naar het begin van de regel.
• CTRL-E	Verplaatst de cursor naar het einde van de regel.
• CTRL-T	Verplaatst de cursor naar het begin van de edit- of regelbuffer.
• CTRL-B	Verplaatst de cursor naar het einde van de edit- of regelbuffer.

Invoegen en wissen

* Return-toets	Maakt een nieuwe regel aan door de cursor en de rest van de tekstregel te verplaatsen naar het begin van de nieuwe regel op het scherm. Als je buiten de editor bent, druk je op de Return-toets om door Logo te laten uitvoeren wat je net hebt getypt.
INS	Maakt een regel vrij op de positie van de cursor, maar verplaatst de cursor niet.
* BS	Doet het teken weg dat links van de cursor staat.
* DEL	Doet het teken weg waar de cursor op staat.
* CTRL-K	Wist de tekst vanaf de cursorpositie tot aan het einde van de regel. Deze tekst wordt in de wisbuffer gezet. Er passen 256 karakters in de wisbuffer.
* CTRL-Y	Voegt op de plaats van de cursor een kopie in van de tekst die in de wisbuffer staat.

De editor verlaten

ESC	Deze letters staan voor **ESC**ape. Hiermee verlaat je de editor en alle proceduredefinities en variabelen die je getypt of veranderd hebt, worden in het werkgeheugen gezet. Als er instructies zijn die niet in een procedure zijn gezet, dan worden die uitgevoerd.
* CTRL-STOP	Hiermee ga je weg uit de editor, terwijl de veranderingen die je hebt aangebracht niet worden ingevoerd. Als je een procedure aan het definiëren was, zal de proceduredefinitie nog hetzelfde zijn als voordat je begon met editen. Buiten de editor kun jë met de toetscombinatie CTRL-STOP alle procedures stoppen.

Opmerking: toetsen of toetscombinaties met een sterretje (*) werken zowel binnen als buiten de Logo-editor.

De Vorm-editor

Je gebruikt de opdracht **ev** (**e**dit **v**orm) om een vorm in de Vorm-editor te zetten, zodat je hem kunt veranderen of een nieuwe vorm kunt aanmaken. De Vorm-editor is een vierkant raster, dat uit 16 x 16 vakken bestaat. De gevulde (zwarte) vakken geven de vorm aan. De cursor (een knipperend wit vierkantje) geeft aan op welke plaats je bezig bent. Gebruik de volgende toetsen of toetscombinaties om vormen aan te maken in de Vorm-editor.

[→]	Verplaatst de cursor een vak naar rechts.
[←]	Verplaatst de cursor een vak naar links.
[↑]	Verplaatst de cursor een vak omhoog.
[↓]	Verplaatst de cursor een vak omlaag.
[Spatiebalk]	Vult een leeg vak of leegt een gevuld vak.
[CTRL]-[K] (of [CLS/HOME])	Doet de hele vorm weg, waardoor een leeg raster overblijft.
[CTRL]-[Y]	Brengt de vorm terug die in het raster stond voordat je er iets aan veranderde.
[ESC]	Hiermee verlaat je de Vorm-editor. De veranderingen blijven bewaard.
[CTRL]-[STOP]	Hiermee ga je uit de Vorm-editor weg zonder dat je de veranderingen laat bewaren.

TN

RI

N+

INT

:N *

Speciale Toetsen

Spatiebalk

Als je op de **spatiebalk** drukt, zet je daarmee een onzichtbaar maar wel heel belangrijk karakter neer, dat een spatie wordt genoemd. Logo gebruikt spaties om woorden van elkaar te scheiden. Als je bijvoorbeeld "**Ditiseenwoord**" typt, ziet Logo dat als een enkel woord, maar "**Dit is een woord**" zijn voor Logo vier woorden.

Hoofdlettertoets of SHIFT-toets

Als je de **hoofdlettertoets** ingedrukt houdt terwijl je op een andere toets drukt, krijgt die toets een andere betekenis voor Logo. Als je een hoofdletter of hoofdletterkarakter wilt typen, moet je altijd eerst de **hoofdlettertoets** ingedrukt houden en daarna de andere toets erbij indrukken.

CTRL-toets

Als je tegelijkertijd op de **CTRL**-toets en een andere toets drukt, krijg je verschillende functies. Als je bijvoorbeeld de **CTRL**-toets ingedrukt houdt terwijl je op de **A** drukt, zal de cursor naar het begin van de regel gaan. Als je een **CTRL**-functie wilt uitvoeren, moet je altijd eerst de **CTRL**-toets ingedrukt houden en daarna de andere toets erbij indrukken.

TAB -toets

Je gebruikt de TAB-toets voor het indelen van procedures. Indelen wil zeggen dat je iets verandert aan de manier waarop de procedure eruit ziet, zodat hij gemakkelijker te lezen en te begrijpen is. Je kunt een lange regel in een aantal korte regels veranderen met behulp van de TAB-toets. Als je op TAB drukt, wordt aan het einde van de schermregel een pijltje neergezet, dat aangeeft dat de tekst op de volgende regel verder gaat, en de cursor gaat naar de volgende regel. De nieuwe regel kan beginnen met een willekeurig aantal spaties, zodat de tekst inspringt. Hieronder zie je een voorbeeld van een ingedeelde procedure:

```
leer test :x :y
als :x > :y                  ➡ Druk op de TAB-toets.
    [drukaf [groter]]        ➡ Inspringen met spaties. Druk op de
                               TAB-toets.
    [drukaf [kleiner]]         Inspringen met spaties.
eind
```


78
657 48
5
6589
2

Speciale Karakters

" (aanhalingstekens)

Aanhalingstekens, ", die je vlak voor een woord gebruikt, geven aan dat het woord gebruikt wordt als zichzelf, en niet als de naam van een procedure of de waarde van een variabele.

Opmerking:
getallen zijn woorden die geen aanhalingstekens nodig hebben.

Voorbeelden:

```
?drukaf "hallo
hallo

?drukaf 5
5
```

: (dubbele punt)

Een dubbele punt, : , die je vlak voor een woord gebruikt, geeft aan dat het woord gezien moet worden als de naam van een variabele. Dit levert de waarde van die variabele op.

Voorbeelden:

```
?maak "paars 13
?drukaf :paars
13

?kleursp :paars
```

[ ] (vierkante haakjes)

Vierkante haakjes , [ ], staan om een lijst heen. Ze kunnen ook bepaalde onderdelen, of elementen, van een lijst bij elkaar houden.

Voorbeelden:

```
?maak "kleuren [rood blauw]
?laatzien :kleuren
[rood blauw]

?maak "kleur "blauw
?(drukaf [Ik heb een] :kleur→
 "huis)
Ik heb een blauw huis
```

Als je een Frans AZERTY-toetsenbord gebruikt, dan gebruik je voor het typen van rechte haken drie toetsen. Als je [of] wilt typen, dan druk je eerst op CODE. Daarna moet je de SHIFT -toets ingedrukt houden en op (of) drukken. Tot slot moet je nog een keer op CODE drukken, zodat je deze toets weer uitschakelt.

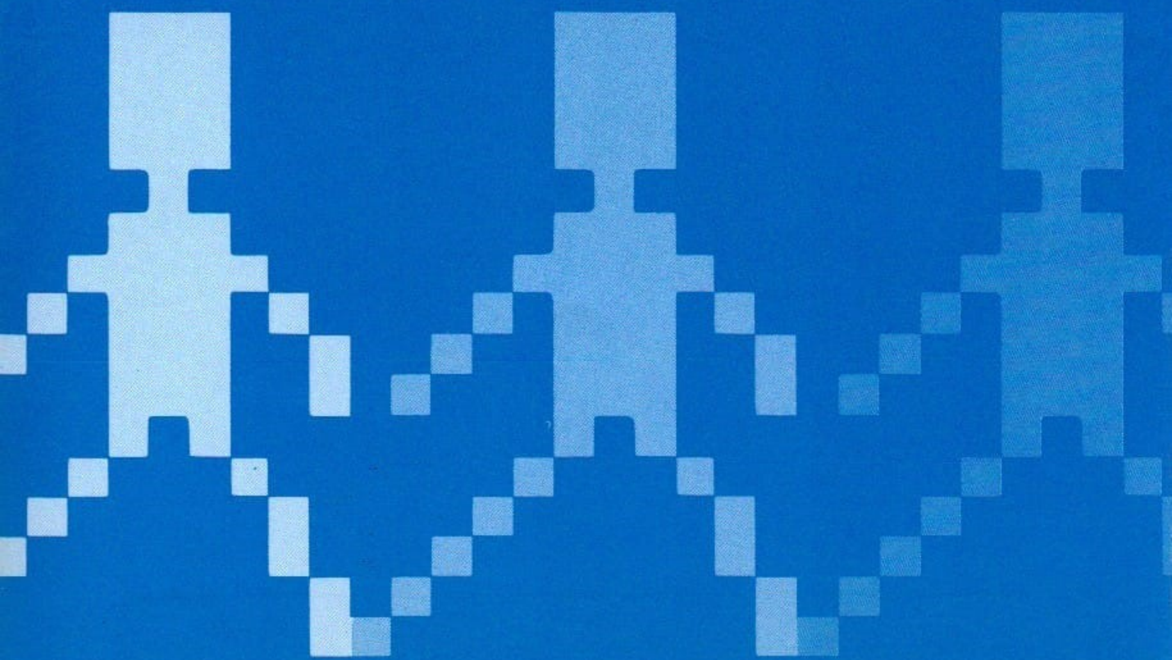
\ (Schuine streep achteruit)

De schuine streep achteruit, \, maakt het mogelijk om een scheidingsteken, bijvoorbeeld een spatie of een rekensymbool, als een normaal karakter te gebruiken. Je moet een \ voor de volgende karakters zetten: [, (,],), +, —, *, /, =, <, > en de \ zelf. Als je een lege spatie wilt afdrukken, kun je dat doen met behulp van de \.

Voorbeelden:

```
?drukaf [985-4482]
985 - 4482

?drukaf [985\-4482]
985-4482
```



Beschrijving van de Primitieve Procedures van MSX-Logo

Dit gedeelte bestaat uit een beschrijving van alle primitieve procedures van MSX-Logo. Na de uitleg over hoe de procedures beschreven worden, komen de volgende onderdelen aan de orde:

Haakjes
Infix-bewerkingen
Speciale Primitieve Procedures
Alfabetische Lijst Van Primitieve Procedures

Hoe We Primitieve Procedures Beschrijven

Bovenaan iedere beschrijving staat de vetgedrukte naam van de primitieve procedure. Als er een verkorte vorm voor de naam bestaat, vind je die erachter. Op dezelfde regel staat vermeld welk soort input nodig is. (De lijst met input-woorden die hier volgt, geeft een definitie van alle input-woorden.) Van elke primitieve procedure wordt aangegeven of het een opdracht of een bewerking is. Daarna volgt een korte uitleg en een voorbeeld waarin de primitieve procedure wordt gebruikt.

Input-woorden

Soort input	Werkelijke input
adres	Een geheel getal van Ø tot en met 65535.
amplitude	Een geheel getal van Ø tot en met 15.
byte	Een geheel getal van Ø tot en met 255.
duur	Een geheel getal van Ø tot en met 255. (5Ø is een seconde.)
filenaam	Een enkel woord waarmee je een file benoemt. Het woord mag voor een schijf niet langer dan 8 karakters en voor een bandje niet langer dan 6 karakters zijn. Je mag geen punten of komma's in de naam zetten.
frequentie	Een geheel getal van 28 tot en met 4Ø95.
gebeurtenisnummer	Een geheel getal van Ø tot en met 5.
getal	Een getal. (Er wordt altijd aangegeven om welk soort getal het gaat.)
instructielijst	Een lijst met procedures die Logo kan uitvoeren.
joysticknummer	Een geheel getal, en wel 1 of 2.
kanaal	Een geheel getal van Ø tot en met 2.
kar	Een letter van het alfabet, een cijfer of een leesteken.
kleurnummer	Een geheel getal van Ø tot en met 15. Logo accepteert hogere getallen, maar deelt het getal dan door 16, waarbij de rest als input wordt gebruikt (bijvoorbeeld 16 = Ø, 17 = 1, 18 = 2).
kolom	Een geheel getal van Ø tot en met 28.
lijst	Informatie die tussen vierkante [ ] haakjes staat.

naam	Een naam van een procedure, een variabele of een property-lijst.
naamlijst	Een lijst met namen van procedures, variábelen of property-lijsten.
nieuwenaam	Een woord dat wordt gebruikt om een procedure te benoemen.
nieuwevormnummer	Een geheel getal van Ø tot en met 59.
object	Een Logo-object (een woord, een lijst of een nummer).
poort	Een getal van Ø tot en met 255.
pred	Een predikaat, dat is een bewerking die ofwel het woord **waar** ofwel het woord **vals** teruggeeft.
prop	Een woord dat een property benoemt.
regel	Een geheel getal van Ø tot en met 23.
schildpadnummer	Een geheel getal van Ø tot en met 29.
schildpadnummerlijst	Een lijst met gehele getallen van Ø tot en met 29.
toon	Een geheel getal van Ø tot en met 31.
vormnummer	Een geheel getal van Ø tot en met 59.
vormspec	Een lijst met 32 getallen die het vormraster voorstellen.
woord	Een opeenvolgende reeks karakters (waar geen spaties bij zijn).
x, y	Een getal dat de x-coördinaat of de y-coördinaat voorstelt.

Een nummerteken (#) wil zeggen dat de primitieve procedure een willekeurig aantal inputs kan hebben. Als je de procedure meer of minder inputs geeft dan hier vermeld staan, moet je de hele uitdrukking tussen haakjes zetten.

Opmerking:
als je een bewerking gebruikt die met de schildpadden te maken heeft, wordt alleen de informatie teruggegeven die van toepassing is op de eerste schildpad uit de groep die door **zeg** worden opgeroepen. Je kunt informatie krijgen over de andere schildpadden die door **zeg** worden opgeroepen, door de opdracht **elk** te gebruiken. Bekijk de voorbeelden die bij **elk** besproken worden.

Haakjes

( )
Deze worden gebruikt om dingen in groepen te plaatsen die Logo normaal niet als één geheel zou zien, of om er de volgorde van het uitvoeren van rekenkundige bewerkingen mee te veranderen. Ook kun je met haakjes bij bepaalde procedures een ander aantal inputs opgeven dan normaal.

Voorbeelden:

```
?drukaf 3 * 5 + 2
17
?drukaf 3 * (5 + 2)
21

?(drukaf "Goede "morgen)
Goede morgen
```

Infix-bewerkingen

Infix-bewerkingen zijn bewerkingen die tussen hun inputs geplaatst worden. Dit zijn allemaal rekenkundige bewerkingen. De bewerking = kan zowel met woorden en lijsten als met getallen worden gebruikt.

getal1 * *getal2* bewerking

Geeft de uitkomst terug van de vermenigvuldiging van *getal1* met *getal2*.

Voorbeeld: `?drukaf 36Ø * 2`

getal1 + *getal2* bewerking

Geeft de uitkomst terug van het optellen van *getal1* en *getal2*.

Voorbeeld: `?drukaf 271,5 + 28,85`

getal1 — *getal2* bewerking

Geeft de uitkomst terug van het aftrekken van *getal 2* van *getal1*.

Voorbeeld: `?drukaf 36Ø - 12Ø`

getal1 / *getal2* bewerking

Geeft de uitkomst terug van *getal1* gedeeld door *getal 2*.

Voorbeeld: `?drukaf 72Ø / 5`

getal1 < *getal2* bewerking

Geeft **waar** terug als *getal1* kleiner is dan *getal2;* geeft in het andere geval **vals** terug.

Voorbeeld: `?laatzien 5 < 4`

object1 = *object2* bewerking

Geeft **waar** terug als *object1* gelijk is aan *object2;* geeft in het andere geval **vals** terug.

Voorbeeld: `?laatzien 1Ø = 5 + 5`

getal1 > *getal2* bewerking

Geeft **waar** terug als *getal1* groter is dan *getal2;* geeft in het andere geval **vals** terug.

Voorbeeld: `?als xcoor > 1ØØ [zetx Ø]`

Speciale Primitieve Procedures

De namen van deze primitieve procedures beginnen met een punt (.), om je te waarschuwen dat het gevaarlijke procedures zijn.

De primitieve procedures **.in** en **.uit** worden gebruikt om te lezen van en te schrijven naar een van de 256 mogelijke input-outputpoorten. Kijk in de technische gegevens van je MSX-computer als je wilt weten welke poorten beschikbaar zijn.

Met **.roep**, **.plaats** en **.controleer** kun je vanuit Logo de assembleertaalprogramma's gebruiken en direct toegang krijgen tot het geheugen. De inputs *adres* en *byte* worden weergegeven in decimale vorm.

.controleer *adres* — bewerking

Geeft de byte terug die opgeslagen is op het machine-adres *adres.* De waarde die teruggegeven wordt, is een geheel getal tussen Ø en 255.

.in *poort* — bewerking

Geeft de byte terug (in decimale vorm) die uit de *poort* gelezen wordt.

.plaats *adres byte* — opdracht

Schrijft de waarde *byte* in het geheugen op het machine-adres *adres.*

.roep *adres* — opdracht

Brengt de besturing over naar een subroutine in machinetaal die begint op het gegeven *adres.*

.uit *poort byte* — opdracht

Stuurt *byte* (in decimale vorm) naar *poort.*

Alfabetische Lijst van Primitieve Procedures

achteruit, at *getal* — opdracht

Verplaatst de schildpad(den) *getal* stappen terug.

Voorbeeld: `?at 4Ø`

afstand [*x y***]** — bewerking

Geeft de afstand terug van de huidige schildpad tot de positie [*x y*].

Voorbeeld: `?drukaf afstand [1Ø Ø]`

allemaal — bewerking

Geeft de lijst van de nummers Ø tot en met 29 terug.

Voorbeeld: `?zeg allemaal drukaf wie`

allesweg — opdracht

Haalt alle procedures, variabelen en property-lijsten uit het werkgeheugen.

Voorbeeld: `?allesweg`

als *pred lijst*
als *pred lijst1 lijst2* — opdracht of bewerking

Als *pred* **waar** teruggeeft, dan wordt *lijst* of *lijst1* uitgevoerd; als *pred* **vals** teruggeeft, dan wordt de optionele *lijst2* uitgevoerd.

Voorbeeld:
```
?als xcoor = Ø [re 18Ø] [re 9→
Ø]
```

arctan *getal* — bewerking

Staat voor **arctan**gens. Geeft de tangens-waarde in graden (-9Ø tot 9Ø) van *getal*.

Voorbeeld: `?drukaf arctan 4`

ascii *kar* — bewerking

Staat voor **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange. Geeft de ASCII-waarde van *kar* terug.

Voorbeeld: `?drukaf ascii "s`

behang — opdracht

Behangt een gedeelte van het scherm met kopieën van de vorm(en) van de huidige schildpad(den).

Voorbeeld: `?behang`

bewaar *filenaam* — opdracht

Bewaart alle procedures, variabelen en property-lijsten die in het werkgeheugen zitten in een schijf-file met de naam *filenaam.*

Voorbeeld: `?bewaar "nieuw`

bewaarpl *filenaam* — opdracht

Staat voor **bewaar pl**aatje. Bewaart de afbeelding op het scherm in een schijf-file met de naam *filenaam.*

Voorbeeld: `?bewaarpl "plaatje`

cbewaar *filenaam* — opdracht

De **c** van deze opdracht staat voor **c**assette. Bewaart alle procedures, variabelen en property-lijsten die in het werkgeheugen zitten in een file met de naam *filenaam* op een cassetteband.
Opmerking: denk eraan dat de knoppen **RECORD** en **PLAY** op de data recorder moeten worden ingedrukt, vóór je **cbewaar** gebruikt.

Voorbeeld: `?cbewaar "mijnfile`

claad *filenaam* — opdracht

De **c** van deze opdracht staat voor **c**assette. Laadt de inhoud van *filenaam* vanaf de cassetteband in het werkgeheugen. De knop **PLAY** op de data recorder moet ingedrukt zijn.
Opmerking: controleer of *filenaam* inderdaad op de cassetteband staat. Anders zal Logo (tevergeefs) blijven zoeken.

Voorbeeld: `?claad "les`

cos *getal* bewerking

Staat voor **cos**inus. Geeft de cosinus van *getal* (in graden).

Voorbeeld: `?drukaf cos 60`

cursor bewerking

Geeft de positie van de cursor terug met twee getallen, het nummer van de kolom en het nummer van de regel.

Voorbeeld: `?drukaf cursor`

da *naam*
da *naamlijst* opdracht

Staat voor **d**ruk **a**f. Drukt alle benoemde procedures af.

Voorbeeld: `?da [poly spi]`

dademn opdracht

Staat voor **d**ruk **a**f **dem**onen. Drukt alle **raak.aan**-demonen en **wanneer**-demonen en hun instructielijsten af.

Voorbeeld: `?dademn`

dafile *filenaam* opdracht

Staat voor **d**ruk **af file**. Drukt de file *filenaam* af.

Voorbeeld: `?dafile "menu`

dalles opdracht

Staat voor **d**ruk **af alles**. Drukt alle procedures, variabelen en property-lijsten af die in het werkgeheugen zitten.

Voorbeeld: `?dalles`

dann opdracht

Staat voor **d**ruk **af n**ame**n**. Drukt alle variabele namen en hun waarden af.

Voorbeeld: `?dann`

daprops opdracht

Staat voor **dr**uk **af prop**ertie**s**. Drukt alle property-lijsten af die in het werkgeheugen zitten.

Voorbeeld: `?daprops`

daps opdracht

Staat voor **d**ruk **af p**rocedure**s**. Drukt alle procedures af die in het werkgeheugen zitten.

Voorbeeld: `?daps`

dats opdracht

Staat voor **d**ruk **af t**itel**s**. Drukt de titels af van alle procedures in het werkgeheugen.

Voorbeeld: `?dats`

definieer *naam lijst* opdracht

Definieert een procedure zonder gebruikmaking van de Editor; wordt gewoonlijk gebruikt binnen een procedure.

Voorbeeld:
```
?definieer "dag [[ ] [drukaf→
"hallo]]
```

definitiep *naam* bewerking

Geeft **waar** terug als *naam* een proceduredefinitie is; geeft in andere gevallen **vals** terug.

Voorbeeld:
```
?als definitiep "tri [da "tri→
]
```

demweg opdracht

Staat voor **dem**onen **weg**. Haalt alle demonen weg (**raak.aan**-botsingen en **wanneer**-gebeurtenissen).

Voorbeeld: `?demweg`

ding *naam* — bewerking

Geeft de inhoud terug van de variabele *naam*.

Voorbeeld: `?drukaf ding "mijnnaam`

drukaf *object* — opdracht

Drukt *object* af op het scherm, gevolgd door een terugloop.

Voorbeelden:
```
?drukaf "hallo
?(drukaf "een [twee drie])
```

edit, ed *naam*
edit *naamlijst* — opdracht

Start de Editor (gelijk aan **leer**, maar **edit** kan ook een lijst bij zich hebben).

Voorbeelden:
```
?edit "poly
?edit [poly spi]
```

ednn — opdracht

Staat voor **ed**it **n**ame**n**. Zet alle variabele namen en hun waarden in de Editor.

Voorbeeld: `?ednn`

eerste *object* — bewerking

Geeft het eerste element van *object* terug.

Voorbeeld: `?drukaf eerste "hallo`

eind — opdracht of bewerking

Geeft het woord **eind** terug. **Eind** wordt gebruikt om een proceduredefinitie die begint met **leer**, te beëindigen.

element *getal object* — bewerking

Geeft het element in *object* terug waarvan de positie in *object* overeenkomt met *getal*.

Voorbeeld:
```
?laatzien element 3 [[een] [t→
wee] [drie]]
```

elk *instructielijst* opdracht

Laat *instructielijst* achter elkaar uitvoeren door elke luisterende schildpad.

Voorbeelden:
```
?zeg [Ø 1 2 3 4] elk [drukaf→
spkleur]
?zeg [1 2 3] elk [ts zetx wie→
 * 2Ø]
```

en *pred1 pred2* bewerking

Geeft **waar** terug als alle inputs **waar** zijn, en geeft **vals** terug als één of meer inputs **vals** zijn.

Voorbeeld:
```
?drukaf en -1Ø < xcoor ycoor →
< 1Ø
```

ev *vormnummer* opdracht

Staat voor **e**dit **v**orm. Zet vorm *vormnummer* in de Vorm-editor.

Voorbeeld:
```
?ev 4
```

fileweg *filenaam* opdracht

Haalt de file *filenaam* weg van de schijf.

Voorbeeld:
```
?fileweg "mijnfile
```

geefterug, gt *object* opdracht

Stopt de huidige procedure en geeft *object* terug. Wordt alleen binnen een procedure gebruikt.

gelijkp *object1 object2* bewerking

Geeft **waar** terug als *object1* en *object2* gelijk zijn; geeft in andere gevallen **vals** terug.

Voorbeeld:
```
?laatzien gelijkp 2 + 2 4
```

geluid *toon amplitude duur* opdracht

Laat een geluid horen met een filterkarakteristiek van *toon*, een sterkte van *amplitude* en een lengte van *duur*.

Voorbeeld:
```
?geluid 1Ø 12 6Ø
```

get *getal* bewerking

Staat voor **ge**heel ge**t**al. Geeft *getal* als een geheel getal terug.

Voorbeeld: ? `drukaf get 3,1415`

getalp *object* bewerking

Geeft **waar** terug als *object* een getal is; geeft anders **vals** terug.

Voorbeeld: ? `drukaf getalp 3,142`

gok *getal* bewerking

Geeft een willekeurig geheel getal terug tussen Ø en *getal* min 1.

Voorbeeld: ? `herhaal 5 [drukaf gok 1Ø]`

gum opdracht

Geeft de huidige schildpad(den) een "**gum**". Alle regels waar de schildpad langs komt, worden weggehaald.

Voorbeeld: ? `gum`

halt opdracht

Laat alle schildpadden stoppen tot een opdracht **start** wordt gegeven.

Voorbeeld: ? `halt wacht 6Ø start`

hap opdracht

Vervangt de vorm van de huidige schildpad(den) door het plaatje dat eronder zit. Als je de opdracht **hap** op een leeg gedeelte van het scherm laat uitvoeren, zal dit de huidige schildpad(den) onzichtbaar maken.

Voorbeeld: ? `hap`

herhaal *getal instructielijst* opdracht

Voert de *intructielijst getal* keren uit.

Voorbeeld: ? `herhaal 3 [vt 5Ø re 12Ø]`

inhoud opdracht

Geeft de namen van alle bestanden op de schijf in de standaardschijfeenheid.

Voorbeeld: `?inhoud`

joy *joysticknummer* bewerking

Geeft een getal tussen Ø en 8 terug, dat de huidige positie van de joystick op *joysticknummer* aangeeft.

Voorbeeld: `?drukaf joy 1`

joyp *joysticknummer* bewerking

Geeft **waar** terug als de joystick op *joysticknummer* niet in het midden staat; geeft anders **vals** terug.

Voorbeeld:
```
?als joyp 1 [richt (joy 1) * →
45]
```

kar *getal*

Staat voor **kar**akter. Geeft het karakter terug waarvan de ASCII-waarde *getal* is.

Voorbeeld: `?drukaf kar 99`

kleur.over bewerking

Geeft het nummer terug van de kleur onder de pen van de schildpad.

Voorbeeld: `?drukaf kleur.over`

kleurscherm *kleurnummer* opdracht

Stelt de achtergrondkleur in op kleur *kleurnummer.*

Voorbeeld: `?kleurscherm 1`

kleursp *kleurnummer* opdracht

Staat voor **kleur s**child**p**ad. Geeft de huidige schildpad(den) kleur *kleurnummer.*

Voorbeeld: `?kleursp 3`

knopp *joysticknummer* — bewerking

Geeft **waar** terug als de knop op *joysticknummer* is ingedrukt; geeft anders **vals** terug.

Voorbeeld: ?`als knopp 1 [vraag 1 [re 3Ø]→`
`]`

kopie — opdracht

Zet een kopie van vorm(en) van de huidige schildpad(den) op het scherm.

Voorbeeld: ?`pn kopie vt 5Ø`

kopiedef *naam nieuwenaam* — opdracht

Staat voor **kopie def**initie. Kopieert de definitie van de procedure *naam* in een nieuwe procedure *nieuwenaam.*

Voorbeeld: ?`kopiedef "oudenaam "nieuwena→`
`am`

kopiev *vormnummer nieuwevormnummer* — opdracht

Staat voor **kopie v**orm. Kopieert de vorm van *vormnummer* in *nieuwevormnummer.*

Voorbeeld: ?`kopiev 5 13`

laad *filenaam* — opdracht

Laadt de inhoud van *filenaam* vanaf schijf in het werkgeheugen.

Voorbeeld: ?`laad "woorden`

laadpl *filenaam* — opdracht

Staat voor **laad pl**aatje. Laadt de afbeelding in de file *filenaam* vanaf de schijf in het werkgeheugen.

Voorbeeld: ?`laadpl "tekening`

laatste *object* — bewerking

Geeft het laatste element van *object* terug.

Voorbeeld: `?drukaf laatste [Ø 1 2 3 4]`

laatzien *object* — opdracht

Drukt *object* af, gevolgd door een terugloop (verwijdert geen haakjes).

Voorbeeld:
```
?laatzien [hallo [goedendag][→
welkom]]
```

leegp *object* — bewerking

Geeft **waar** terug als *object* leeg is; geeft anders **vals** terug.

Voorbeelden:
```
?laatzien leegp [1 2]
?laatzien leegp [ ]
```

leer
leer *naam* — opdracht

Start de Editor.

Voorbeeld: `?leer vierkant`

lijst *object1 object2* — bewerking

Geeft een lijst terug die samengesteld is uit zijn inputs. Zie ook zin.

Voorbeeld:
```
?laatzien lijst "een [twee dr→
ie]
```

lijstp *object* — bewerking

Geeft **waar** terug als *object* een lijst is; geeft anders **vals** terug.

Voorbeeld: `?drukaf lijstp [a b c]`

links, li *getal* — opdracht

Draait de luisterende schildpad(den) *getal* graden naar links.

Voorbeeld: `?li 9Ø`

lk bewerking

Staat voor **l**ees **k**arakter. Leest één karakter van het toetsenbord en geeft het terug.

Voorbeeld: `?drukaf zin [die je typte] lk`

ll bewerking

Staat voor **l**ees **l**ijst. Geeft de lijst terug die vanaf het toetsenbord is ingevoerd. (Met de **Return**-toets geef je het einde van de lijst aan.)

Voorbeeld: `?drukaf tel ll`

lw opdracht

Staat voor lijnen weg. Haalt de lijnen weg van het scherm en brengt de huidige schildpad(den) thuis.

Voorbeeld: `?lw`

maak *naam object* opdracht

Geeft een *naam* aan *object*. Hiermee wordt een variabele gemaakt die blijft bestaan tot ze wordt veranderd of weggehaald.

Voorbeeld: `?maak "mijnnaam "rene`

mineerste, me *object* bewerking

Geeft alles behalve het eerste element van *object* terug.

Voorbeeld: `?drukaf me [hond kat muis]`

minlaatst, ml *object* bewerking

Geeft alles behalve het laatste element van *object* terug.

Voorbeeld:
```
?drukaf ml [personal computer→
]
```

ms opdracht

Staat voor **m**aak **s**choon. Haalt alle tekeningen weg van het scherm en zet de schildpad(den) terug in de beginstand. Demonen worden ook weggehaald. De tekst komt weer in zijn oorspronkelijke kleur (wit).

Voorbeeld: `?ms`

naamp *naam* bewerking

Geeft **waar** terug als *naam* de naam is van een woord of een lijst (een variabele); geeft anders **vals** terug.

Voorbeeld: `?drukaf naamp "mijnnaam`

naar [*x y***]** bewerking

Geeft de richting terug vanaf de huidige schildpad naar de positie [*x y*].

Voorbeeld: `?richt naar [20 20]`

neemv *vormnummer vormspec* opdracht

Staat voor **neem** vorm. Geeft aan vorm *vormnummer* de vorm die gedefinieerd wordt door *vormspec*. Zie ook **pakv.**

Voorbeeld: `?neemv 12 pakv 3`

niet *pred* bewerking

Geeft **waar** terug als *pred* **vals** is of geeft **vals** terug als *pred* **waar** is.

Voorbeeld:
```
?als niet 0 = xcoor [zetx -xc→
oor]
```

nnweg opdracht

Staat voor **n**ame**n** **weg.** Haalt alle variabele namen weg uit het werkgeheugen.

Voorbeeld: `?nnweg`

noem *object naam* opdracht

Geeft een *naam* aan *object.* Hiermee wordt een variabele gemaakt die blijft bestaan tot ze wordt veranderd of weggehaald. **Naam** is hetzelfde als **maak**, maar met omgekeerde inputs.

Voorbeeld: `?noem 1,25 "bedrag`

nweg *naam*
nweg *naamlijst* — opdracht

Staat voor **n**aam **weg.** Haalt de benoemde variabele(n) weg uit het werkgeheugen.

Voorbeeld: `?nweg "var.naam`

of *pred1 pred2* — bewerking

Geeft **waar** terug als één of meer inputs **waar** zijn, of geeft **vals** terug als alle inputs **vals** zijn.

Voorbeeld:
```
?als of xcoor > -5Ø 5Ø < xcoo→
r [drukaf "buiten]
```

pakprop *naam prop* — bewerking

Staat voor **pak prop**erty. Geeft de waarde terug van de property *prop* uit de property-lijst *naam.*

Voorbeeld:
```
?laatzien pakprop "schildpad1→
 "kleur
```

pakv *vormnummer* — bewerking

Staat voor **pak v**orm. Geeft een lijst terug van 32 nummers die de schildpadvorm *vormnummer* weergeven. Zie ook **neemv.**

Voorbeeld: `?maak "hart pakv 1`

penneer, pn — opdracht

Zet de pen van de huidige schildpad(den) omlaag, zodat ze lijnen kunnen tekenen.

Voorbeeld: `?pn`

penop, po — opdracht

Houdt de pen van de huidige schildpad(den) omhoog.

Voorbeeld: `?po`

pk bewerking

Staat voor **p**en**k**leur. Geeft de penkleur van de huidige schildpad(den) terug.

Voorbeeld: `?drukaf pk`

pos bewerking

Staat voor **pos**itie. Geeft de schermpositie van de huidige schildpad terug met twee getallen (de x-coördinaat en de y-coördinaat).

Voorbeeld: `?drukaf pos`

primitiefp *naam* bewerking

Geeft **waar** terug als *naam* een primitieve procedure is; geeft anders **vals** terug.

Voorbeeld: `?drukaf primitiefp "vt`

primitieven opdracht

Geeft een lijst met alle namen van de primitieve procedures van MSX-Logo.

Voorbeeld: `?primitieven`

printer opdracht

Opent het output-kanaal naar de printer.

Voorbeeld: `?printer`
Voorbeeld: `?drukaf "hallo printeruit`

printeruit opdracht

Sluit het output-kanaal naar de printer.

Voorbeeld: `?printeruit`

produkt *getal1 getal2* bewerking

Geeft het produkt (de vermenigvuldiging) terug van de input-getallen.

Voorbeeld: `?drukaf produkt 3 5`

proplijst *naam* bewerking

Staat voor **prop**erty-**lijst**. Geeft de property-lijst terug die bij *naam* hoort. Property-lijsten worden gemaakt met **zetprop.**

Voorbeeld: ?`drukaf proplijst "schildpad Ø`

propsweg opdracht

Staat voor **prop**ertie**s weg**. Haalt alle property-lijsten weg uit het werkgeheugen.

Voorbeeld: ?`propsweg`

propweg *naam prop* opdracht

Staat voor **prop**erty **weg**. Verwijdert de property *prop* en zijn waarde uit de property-lijst *naam.*

Voorbeeld: ?`propweg "schildpad2 "vorm`

psweg opdracht

Staat voor **p**rocedure**s weg**. Haalt alle procedures weg uit het werkgeheugen.

Voorbeeld: ?`psweg`

punt [*x y***]** opdracht

Zet een punt op positie [*x y*] op het beeldscherm.

Voorbeeld: ?`punt [2Ø 5Ø]`

px opdracht

Zet de pen van de schildpad(den) omlaag. Er kunnen lijnen worden getekend, tenzij de pen over een al getekende lijn komt. In dat geval wordt de bestaande lijn weggehaald.

Voorbeeld: ?`px`

quotient *getal1 getal2* bewerking

Geeft de uitkomst terug van *getal1* gedeeld door *getal2.*

Voorbeeld: ?`drukaf quotient 6 3`

raak.aan *schildpadnummer1 schildpadnummer2 instructielijst* — opdracht

Zet een demoon op wacht die controleert of *schildpadnummer1* en *schildpadnummer2* tegen elkaar botsen. Wanneer dat gebeurt, wordt *instructielijst* uitgevoerd. Voor het weghalen van demonen, gebruik je de opdracht **demweg.**

Voorbeeld: ?`raak.aan Ø 1 [re 9Ø]`

raam — opdracht

Roept een modus op die van het scherm een raam maakt op een veel groter gebied dan het scherm alleen. Zie ook **wikkel.**

Voorbeeld: ?`ss raam vt 5ØØ`

rechts, re *getal* — opdracht

Draait de luisterende schildpad(den) *getal* graden naar rechts.

Voorbeeld: ?`re 9Ø`

rest *getal1 getal2* — bewerking

Geeft de rest van *getal1* gedeeld door *getal2.*

Voorbeeld: ?`drukaf rest 1Ø 3`

richt *getal* — opdracht

Stelt de richting van de huidige schildpad(den) in op *getal* graden.

Voorbeeld: ?`richt 27Ø`

richting — bewerking

Geeft de richting van de schildpad terug in graden.

Voorbeeld: ?`richt 18Ø + richting`

rondaf *getal* — bewerking

Geeft de afgeronde, gehele waarde van *getal.*

Voorbeeld: ?`drukaf rondaf 3,7`

ruimop opdracht

Ruimt het geheugen van de computer op door loze geheugenruimte te verzamelen. Beschikbare knooppunten en variabelen-ruimte worden vrijgemaakt.

Voorbeeld: `?ruimop`

schermr bewerking

Geeft de aspect ratio van het scherm terug. De aspect ratio is een getal dat uitdrukt hoe de grootte van een verticale stap (van de schildpad) op het scherm zich verhoudt tot de grootte van een horizontale stap op het scherm.

Voorbeeld: `?drukaf schermr`

schoon opdracht

Haalt alle tekeningen weg van het scherm, maar laat de schildpad op zijn plaats staan.

Voorbeeld: `?vt 5Ø schoon`

sin *getal* bewerking

Staat voor **sin**us. Geeft de sinus van *getal* (in graden).

Voorbeeld: `?drukaf sin 3Ø`

sk opdracht

Staat voor **s**cherm**k**leur. Geeft het kleurnummer terug van de achtergrond op het scherm.

Voorbeeld: `?drukaf sk`

som *getal1 getal2* bewerking

Geeft de som (de optelling) terug van de input-getallen.

Voorbeeld: `?drukaf som 6 4`

spkleur bewerking

Staat voor **s**child**p**ad**kleur**. Geeft het kleurnummer terug van de huidige schildpad.

Voorbeeld: ?`drukaf spkleur`

ss opdracht

Staat voor **s**cherm **s**choon. Haalt tekst en tekeningen weg van het scherm en zet de schildpad(den) midden op het scherm. Ze staan stil en "kijken" naar boven. De opdracht **ss** is hetzelfde als **schoon po zetvaart Ø zetpos [Ø Ø] richt Ø pn tw.**

Voorbeeld: ?`ss`

start opdracht

De opdracht volgt na **halt** en laat de schildpad verder lopen met de ingestelde snelheid.

Voorbeeld: ?`zetvaart 1Ø`
Voorbeeld: ?`halt wacht 6Ø start`

stelr *getal* opdracht

Staat voor **stel r**atio. Verandert de grootte van de verticale stappen (van de schildpad) op het scherm in verhouding tot de grootte van de horizontale stappen op het scherm. De normale instelling van de ratio is 1,375.

Voorbeeld: ?`stelr ,7`

stop opdracht

Stopt de huidige procedure. Wordt alleen binnen een procedure gebruikt.

sw opdracht

Staat voor **s**childpad **w**eg. Maakt de huidige schildpad(den) onzichtbaar.

Voorbeeld: ?`sw`

tekst *naam* — bewerking

Geeft de definitie van de procedure *naam* terug als een lijst.

Voorbeeld: ?`laatzien tekst "vierkant`

tel *object* — bewerking

Geeft het aantal elementen in *object* terug.

Voorbeeld: ?`drukaf tel [23 45 15 72 98]`
Voorbeeld: ?`drukaf tel "Nellie`

tezienp — bewerking

Geeft **waar** terug als de huidige schildpad zichtbaar is (na een opdracht **ts**); geeft anders **vals** terug.

Voorbeeld: ?`drukaf tezienp`

thuis — opdracht

Stopt de huidige schildpad(den) en zet ze terug in het midden van het scherm op positie [Ø Ø] en met richting Ø.

Voorbeeld: ?`vraag Ø thuis`

toetsp — bewerking

Geeft **waar** terug als een toets wel ingedrukt maar nog niet gelezen is; geeft anders **vals** terug.

Voorbeeld: ?`laatzien toetsp`

toon *kanaal frequentie amplitude duur* — opdracht

Laat een toon horen op *kanaal*, met een toonhoogte van *frequentie*, een sterkte van *amplitude* en een lengte van *duur*.

Voorbeeld: ?`toon 1 44Ø 1Ø 6Ø`

topnivo opdracht

Onderbreekt de uitvoering van alle procedures en keert terug naar het topnivo. Als de opdracht op het topnivo wordt gebruikt, stopt Logo de regel die wordt uitgevoerd.

Voorbeeld: `?drukaf 5 topnivo drukaf 6`

ts opdracht

Staat voor **t**oon **s**childpad. Maakt de huidige schildpad(den) zichtbaar.

Voorbeeld: `?ts`

tw opdracht

Staat voor **t**ekst **w**eg. Haalt de tekst weg van het scherm.

Voorbeeld: `?tw`

typ *object* opdracht

Zet *object* op het scherm zonder terugloop.

Voorbeeld: `?typ "hallo`

vaart bewerking

Geeft de snelheid van de huidige schildpad terug.

Voorbeeld: `?drukaf vaart`

vals bewerking

Geeft het speciale woord **vals** terug.

Voorbeeld: `?laatzien vals`

verkleur *kleurnummer1 kleurnummer2* opdracht

Verandert de tekeningen en de tekst op het scherm van *kleurnummer1* in *kleurnummer2*.

Voorbeeld: `?verkleur 15 7`

verschil *getal1 getal2* — bewerking

Geeft de uitkomst van *getal1* min *getal2*.

Voorbeeld: ?`drukaf verschil 10 5`

versie — opdracht

Geeft het codenummer van de gebruikte versie van Logo.

Voorbeeld: ?`versie`

voeruit *lijst* — opdracht of bewerking

Voert *lijst* uit. Als *lijst* een bewerking is, dan zal **voeruit** alles teruggeven wat de lijst teruggeeft.

Voorbeeld:
```
?voeruit element 2 [[drukaf "→
hallo] [drukaf "kees] [drukaf→
 "dag]]
```

vooruit, vt *getal* — opdracht

Verplaatst de huidige schildpad(den) *getal* stappen vooruit.

Voorbeeld: ?`vt 50`

vormn — bewerking

Geeft het vormnummer terug van de huidige schildpad.

Voorbeeld: ?`drukaf vormn`

vraag *schildpadnummer lijst*
vraag *schildpadnummerlijst lijst* — opdracht of bewerking

Laat *schildpadnummer* of *schildpadnummerlijst lijst* uitvoeren, maar verandert niet de huidige schildpadlijst (opgeroepen door **zeg**).

Voorbeeld: ?`vraag 5 [ts thuis at 30]`

vul — opdracht

Vult een ingesloten gebied op het scherm (zoals een kader) met de kleur(en) van de pen(nen) van de huidige schildpad(den).

Voorbeeld: ?`pn vul`

waar bewerking

Geeft het speciale woord **waar** terug.

Voorbeeld: ?`laatzien waar`

wacht *getal* opdracht

Onderbreekt een bewerking voor de duur van *getal* 5Østen van een seconde.

Voorbeeld:
```
?herhaal 5 [typ "hallo wacht →
5Ø]
```

wanneer *gebeurtenisnummer instructielijst* opdracht

Laat een **wanneer**-demoon wachten, die controleert of *gebeurtenisnummer* voorkomt. Zodra het nummer voorkomt, wordt *instructielijst* uitgevoerd. Zie ook de Gebeurtenissentabel in "Kleur en Vorm in MSX-Logo".

Voorbeeld: ?`wanneer Ø [vt 1 re 1]`

weg *naam* opdracht
weg *naamlijst*

Haalt de opgegeven procedure(s) weg uit het werkgeheugen.

Voorbeeld: ?`weg "poly`

wie bewerking

Geeft de lijst terug van de huidige schildpad(den).

Voorbeeld: ?`drukaf wie`

wikkel opdracht

Roept een modus op waarin de schildpad zich als het ware om het scherm heen wikkelt. Als hij zover moet lopen dat hij het zichtbare gebied van het scherm dreigt te verlaten, komt hij aan de andere kant weer tevoorschijn. Dit is de standaard wanneer je de computer aanzet en Logo start. De opdracht **wikkel** maakt het scherm schoon. Zie ook **raam**.

Voorbeeld: ?`wikkel vt 5ØØ`

woord *woord1 woord2* — bewerking

Geeft de inputs terug als één woord, zonder spaties.

Voorbeeld: `?drukaf woord "maan "landscha→`
`p`

woordp *object* — bewerking

Geeft **waar** terug als *object* een woord is; geeft anders **vals** terug.

Voorbeeld: `?drukaf woordp "judith`

wortel *getal* — bewerking

Geeft de wortel van *getal* terug.

Voorbeeld: `?drukaf wortel 4`

wr — bewerking

Staat voor **w**erk**r**uimte. Geeft de beschikbare hoeveelheid werkgeheugen.

Voorbeeld: `?drukaf wr`

xcoor — bewerking

Staat voor **x-coör**dinaat. Geeft de x-coördinaat van de huidige schildpad terug.

Voorbeeld: `?drukaf xcoor`

xvaart — bewerking

Geeft de horizontale component terug van de snelheid van de huidige schildpad.

Voorbeeld: `?drukaf xvaart`

ycoor — bewerking

Staat voor **y-coör**dinaat. Geeft de y-coördinaat van de huidige schildpad terug.

Voorbeeld: `?zety (ycoor + 3Ø)`

yvaart bewerking

Geeft de verticale component terug van de snelheid van de huidige schildpad.

Voorbeeld: ?zetyv -(yvaart)

zeg *schildpadnummer*
zeg *schildpadnummerlijst* opdracht

Maakt van *schildpadnummer* of *schildpadnummerlijst* de huidige schildpad(den).

Voorbeeld: ?zeg [Ø 6 21]

zelfdegok opdracht

Genereert dezelfde reeks willekeurige getallen.

Voorbeeld: ?herhaal 5 [zelfdegok drukaf →
gok 1Ø]

zetcursor [*kolom regel***]** opdracht

Zet de cursor op tekstpositie [*kolom regel*].

Voorbeeld: ?zetcursor [1Ø 1Ø]

zete *object lijst* bewerking

Staat voor **zet e**erste. Geeft een *lijst* terug met *object* als het eerste element.

Voorbeeld: ?drukaf zete 1 [2 3]

zetl *object lijst* bewerking

Staat voor **zet l**aatste. Geeft een *lijst* terug met *object* als het laatste element. Zie ook **zete.**

Voorbeeld: ?laatzien zetl "een [2 3]

zetpk *kleurnummer*

Staat voor **zet** **pen**k**l**eur. Stelt de kleur van de pen(nen) van de huidige schildpad(den) in op *kleurnummer.*

Voorbeeld: ? `zetpk 5`

zetpos [*x y***]** opdracht

Staat voor **zet** **pos**itie. Zet de huidige schildpad(den) op de schermpositie [*x y*].

Voorbeeld: ? `zetpos [2Ø -65]`

zetprop *naam prop object* opdracht

Staat voor **zet** **prop**erty. Geeft aan *naam* de property *prop* met *object* als de waarde.

Voorbeeld: ? `zetprop "schildpad Ø "vorm 8`

zettekst *regel* opdracht

Zet de bovenste regel voor tekst op *regel* (Ø bovenkant scherm, 23 onderkant scherm).

Voorbeeld: ? `zettekst 18`

zetv *vormnummer* opdracht

Staat voor **zet** **v**orm. Zet de vorm van de huidige schildpad(den) op *vormnummer* (Ø tot 59).

Voorbeeld: ? `zetv 1Ø`

zetvaart *getal* opdracht

Stelt de snelheid van de huidige schildpad(den) in op *getal* (van -128 tot 128).

Voorbeeld: ? `zetvaart 3Ø`

zetx *x* — opdracht

Zet de huidige schildpad(den) op de x-coördinaat *x*.

Voorbeeld: ?`zetx 90`

zetxv *getal* — opdracht

Staat voor **zet x v**aart. Stelt de x-component van de snelheid van de huidige schildpad(den) in op *getal*.

Voorbeeld: ?`zetxv 0`

zety *y* — opdracht

Zet de huidige schildpad(den) op de y-coördinaat *y*.

Voorbeeld: ?`zety -65`

zetyv *getal* — opdracht

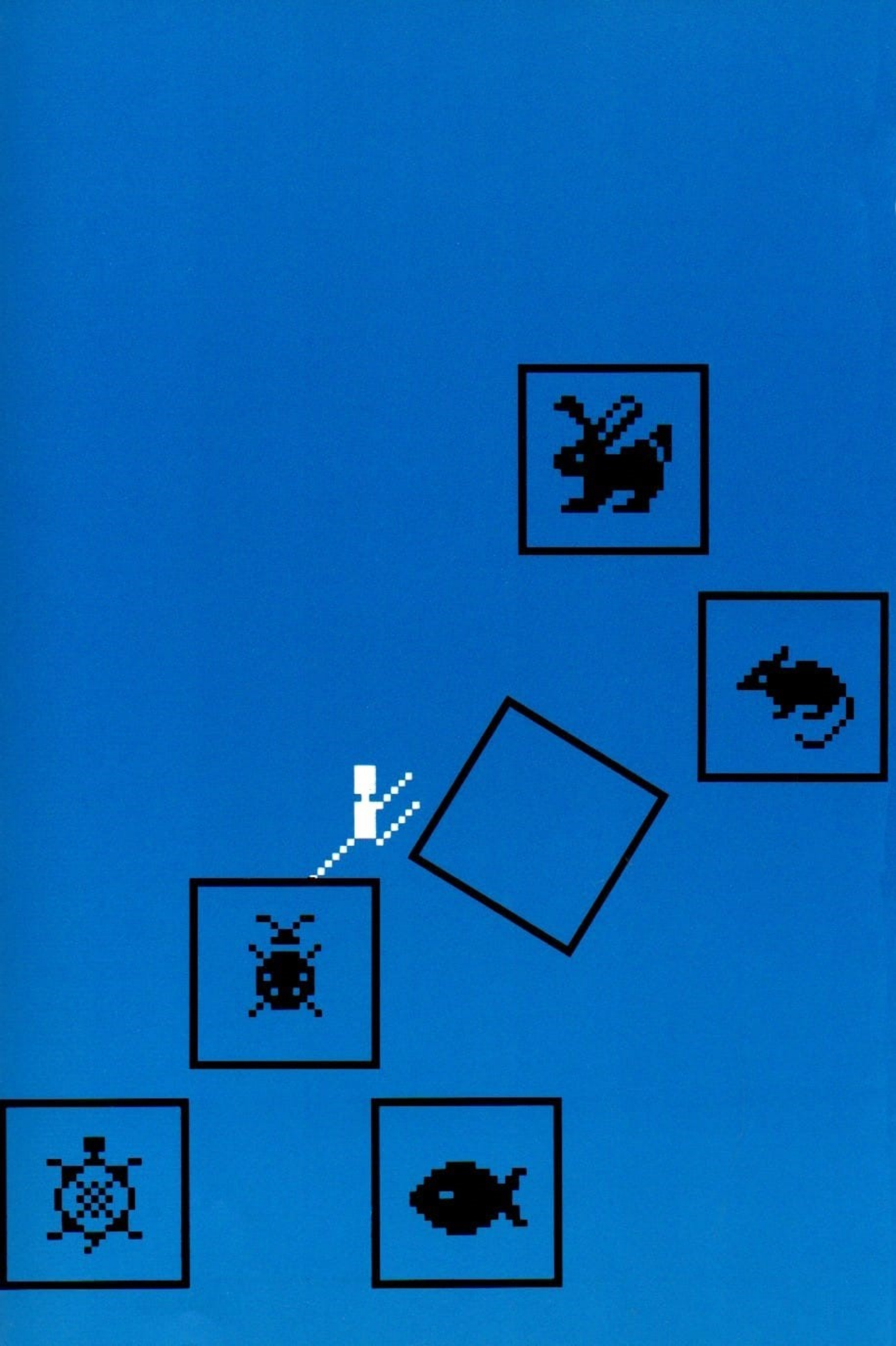
Staat voor **zet y v**aart. Stelt de y-component van de snelheid van de huidige schildpad(den) in op *getal*.

Voorbeeld: ?`zetyv vraag 0 [yvaart]`

**zin** *object1 object2* — bewerking

Maakt van zijn inputs één lijst, en verwijdert daarbij de haakjes. Zie ook **lijst.**

Voorbeeld:
```
?laatzien (zin "een [twee dri→
e 4])
```



MSX-Logo
Primitieve Procedures
Ingedeeld in Functiegroepen

In dit hoofdstuk vind je een overzicht van de primitieve procedures, ingedeeld naar de soort functie of taak die ze uitvoeren. Deze lijst zal je van pas komen als je een primitieve procedure zoekt om een specifieke functie uit te voeren, bijvoorbeeld het maken van bewegende computerplaatjes (sprites) of muziek. De procedures in elke groep staan in alfabetische volgorde.

PLAATJES

achteruit, at *getal*
afstand [*x y***]**
gum
kleurscherm *kleurnummer*
links, li *getal*
lw
naar [*x y***]**
penneer, pn
penop, po
pk
pos
punt [*x y***]**
px
raam
rechts, re *getal*
richt *getal*
richting
schermr
schoon
sk
ss
stelr *getal*
sw
tezienp
thuis
ts
vooruit, vt *getal*
wikkel
xcoor
ycoor
zetpk *kleurnummer*
zetpos [*x y***]**
zetx *x*
zety *y*

WOORDEN EN LIJSTEN

ascii *kar*
eerste
element *getal object*
gelijkp *object1 object2*
getalp *object*
kar *getal*
laatste *object*
leegp *object*
lijst *object1 object2*
lijstp *object*
mineerste, me *object*
minlaatste, ml *object*
tel *object*
woord *woord1 woord2*
woordp *object*
zete *object lijst*
zetl *object lijst*
zin *object1 object2*

VARIABELEN

ding *naam*
maak *naam object*
noem *object naam*
naamp *naam*

REKENKUNDIGE BEWERKINGEN

arctan *getal*
cos *getal*
get *getal*
gok *getal*
produkt *getal1 getal2*
quotient *getal1 getal2*
rest *getal1 getal2*
rondaf *getal*
sin *getal*
som *getal1 getal2*
verschil *getal1 getal2*
wortel *getal*
zelfdegok
getal1 + getal2
getal1 — getal2
getal1 / getal2
*getal1 * getal2*
object1 = object2
getal1 < getal2
getal1 > getal2

LOGISCHE BEWERKINGEN

en *pred1 pred2*
niet *pred*
of *pred1 pred2*

WERKGEHEUGEN

allesweg
da *naam(lijst)*
dalles
dann
daps
dats
nnweg
nweg *naam(lijst)*
primitieven
psweg
ruimop
weg *naam(lijst)*
wr

DEFINIEREN EN EDITEN

definieer *naam lijst*
definitiep *naam*
edit, ed *naam(lijst)*
ednn
eind
kopiedef *naam nieuwenaam*
leer *(naam)*
primitiefp *naam*
tekst *naam*

CONDITIES EN BESTURINGSCONTROLE

als *pred lijst1 (lijst2)*
geefterug, gt *object*
herhaal *getal instructielijst*
stop
topnivo
vals
voeruit *lijst*
waar

SCHERM, TOETSENBORD EN PRINTER

cursor
drukaf *object*
laatzien *object*
lk
ll
printer
printeruit
toetsp
tw
typ *object*
wacht *getal*
zetcursor [*kolom regel***]**
zettekst *regel*

PROPERTY-LIJSTEN

daprops
pakprop *naam prop*
proplijst *naam*
propsweg
propweg *naam prop*
zetprop *naam prop object*

SPRITES

allemaal
behang
dademn
demweg
elk *instructielijst*
ev *vormnummer*
halt
hap
joy *joysticknummer*
joyp *joysticknummer*
kleur.over
kleursp *kleurnummer*
knopp *joysticknummer*
kopie
kopiev *vormnummer nieuwevormnummer*
ms
neemv *vormnummer vormspec*
pakv *vormnummer*
raak.aan *schildpadnummer1 schildpadnummer2 instructielijst*
spkleur
start
vaart
verkleur *kleurnummer1 kleurnummer2*
vormn
vraag *schildpadnummer(lijst) lijst*
vul
wanneer *gebeurtenisnummer instructielijst*
wie
xvaart
yvaart
zeg *schildpadnummer(lijst)*
zetv *vormnummer*
zetvaart *getal*
zetxv *getal*
zetyv *getal*

MUZIEK

geluid *toon amplitude duur*
toon *kanaal frequentie amplitude duur*

BESTANDEN BEHEREN

(alleen voor machines met een 64K-geheugen en schijfeenheden)

bewaar *filenaam*
bewaarpl *filenaam*
dafile *filenaam*
fileweg *filenaam*
inhoud
laad *filenaam*
laadpl *filenaam*

LADEN EN BEWAREN OP CASSETTEBAND

cbewaar *filenaam*
claad *filenaam*

SPECIALE PRIMITIEVEN

.controleer *adres*
.in *poort*
.plaats *adres byte*
.roep *adres*
.uit *poort byte*
versie

Index

W

X

Y

Z

Beperkte aansprakelijkheid

Philips aanvaardt voor dit programma geen aansprakelijkheid. De gebruiker draagt dientengevolge het volledige risico met betrekking tot werking en resultaten van het programma. Mocht het programma ondeugdelijk zijn, dan komen de hieruit voortvloeiende kosten van service, reparatie of correctie volledig voor rekening van de gebruiker (en niet van Philips of de dealers van Philips). Voorts accepteert Philips geen aansprakelijkheid en biedt geen garantie voor het programma met betrekking tot deugdelijkheid, juistheid, betrouwbaarheid, actualiteit of anderszins; de gebruiker gebruikt het programma volledig op eigen risico.

Bovenstaande aansprakelijkheid is de enig bestaande, uitdrukkelijk dan wel geïmpliceerd, met inbegrip van maar niet beperkt tot verkoopbaarheid en deugdelijkheid van het programma voor een speciaal doel. Deze aansprakelijkheid geeft de gebruiker bepaalde rechten; daarnaast kan de gebruiker rechten hebben die van land tot land verschillen.

Noch Philips noch derden die betrokken zijn geweest bij de ontwikkeling, produktie en levering van dit programma, kunnen aansprakelijk worden gesteld voor enige directe, indirecte, resulterende of incidentele schade door het gebruik, als gevolg van het gebruik of door de onmogelijkheid het produkt te gebruiken. Dit geldt ook wanneer Philips vooraf in kennis is gesteld van de mogelijkheid van genoemde schade of schadeclaims. In sommige landen geldt geen uitsluiting van aansprakelijkheid of beperking van aansprakelijkheid voor resulterende of incidentele schade; in die landen is bovenstaande beperking niet geldig.

PHILIPS

PHILIPS MSX-LOGO

MSX-LOGO werd ontwikkeld door Dr. Seymour Papert, een wiskundige, in nauwe samenwerking met de inmiddels overleden Zwitserse kinder-psycholoog Jean Piaget. Die combinatie geeft een afdoende verklaring waarom LOGO kinderlijk eenvoudig is om te leren zonder kinderachtig te zijn wat de mogelijkheden betreft! Dat geldt zeker voor de combinatie van LOGO met MSX, want beide zijn vooral grafisch erg sterk en openen interessante mogelijkheden voor muziek en bewegende beelden.
MSX-LOGO is zeer veelzijdig. De gebezigde uitdrukkingen zijn in uw eigen landstaal; ze zijn logisch en daardoor erg makkelijk te begrijpen, te onthouden en gebruiken zodat ze ideaal zijn voor educatieve doeleinden. Zoals Dr. Papert zelf zegt: 'LOGO kent geen drempel en geen plafond. Het is verrassend makkelijk te leren, maar toch een interessante uitdaging voor ervaren programmeurs.'

LOGO werd in 1980 in Amerika geïntroduceerd en is in recordtijd de op een na meest populaire programmeertaal geworden. De afgelopen twee jaar werd het uitgeroepen tot 'Educatief Software van het Jaar'!

Met LOGO bevestigen wetenschappers en onderwijsdeskundigen het feit, dat de computer het stadium bereikt heeft waarin het niet meer alleen het exclusieve territorium is van specialisten!

In deze makkelijk leesbare, Nederlandstalige handleiding vindt u alle informatie die u nodig hebt om uw eigen programma's in MSX-LOGO te schrijven. Het programma zelf is opgeslagen in een niet-uitwisbare ROM-cassette, direct klaar voor gebruik.

LCSI
Logo Computer Systems Inc.
Montreal, Canada, H8T1A1

VG 8103/23 - 3122 285 70701 Ned.

Printed in Holland